



**Australian Government**  
**Department of Defence**  
Defence Science and  
Technology Organisation

# Wideband High-Resolution Filterbank Firmware: Acceptance Testing and Excision Filtering

*K. L. Harman and C. Potter*

**Command, Control, Communications and Intelligence Division**  
Defence Science and Technology Organisation

DSTO-TR-2092

## **ABSTRACT**

A filterbank is a signal processing tool that can facilitate manipulation of signals in the frequency domain. Under contract to DSTO, RF Engines Limited in the United Kingdom developed a 512-bin and a 4096-bin filterbank as firmware cores, capable of real-time digital processing at complex sample rates of 200MSPS (390kHz per bin) and 100MSPS (24kHz per bin) respectively. These cores will facilitate research and development into high-performance adaptive filtering in support of Defence wireless communications tasks. In this document, the integration of the cores into an existing platform and subsequent acceptance testing are reported, confirming that the cores do satisfy their design specification. Then the application of the cores to front-end excision filtering in a direct-sequence spread-spectrum receiver is investigated, and demonstrates a highly effective performance enhancement.

## **RELEASE LIMITATION**

*Approved for public release*

*Published by*

*Command, Control, Communications and Intelligence Division  
DSTO Defence Science and Technology Organisation  
PO Box 1500  
Edinburgh South Australia 5111 Australia*

*Telephone: (08) 8259 5555  
Fax: (08) 8259 6567*

*© Commonwealth of Australia 2008  
AR-014-084  
January 2008*

**APPROVED FOR PUBLIC RELEASE**

# Wideband High-Resolution Filterbank Firmware: Acceptance Testing and Excision Filtering

## Executive Summary

An aspect of DSTO support to Defence wireless communications tasks is the provision of customised digital transceiver solutions that facilitate embedded signal processing. These signal processing algorithms can be tailored to optimise link quality.

A filterbank is a signal processing tool that can facilitate manipulation of signals in the frequency domain in a computationally efficient manner, for real-time and wideband solutions to tasks such as spectrum analysis, filtering and equalisation. In a filtering application a signal is input to a filterbank analyser in time-domain digital form, converted to an approximate discrete frequency domain representation, manipulated by a set of user-defined weights, then reconstructed into the time domain by a filterbank synthesiser with minimal unintentional alteration. This transparent, in-line method of insertion gives it general applicability in a range of processing applications.

RF Engines Limited, in the United Kingdom, was selected based on their renowned experience in implementing field-programmable gate array (FPGA) based filterbanks to produce two variants of a filterbank suitable for integration into an FPGA-based digital receiver. These cores were to achieve the highest bandwidth and greatest frequency resolution attainable within the constraints of the FPGA technology, and would thereby deliver a leading edge capability for DSTO's receiver platform. Under contract to DSTO, a 200MSPS, 512-bin (390 kHz per bin) intellectual property (IP) core and a 100MSPS, 4096-bin (24 kHz per bin) IP core were produced, both capable of real-time continuous operation with complex time domain input and output formats. Each was specified to have a time-domain signal reconstruction error of no more than 1%.

DSTO undertook integration of the cores into the existing receiver platform, followed by acceptance testing to validate core functionality and compliance. These processes, which are documented herein, resulted in acceptance of the cores with only minor concessions against the design specifications, and availability of the cores for research and development.

Application against narrowband interference in a broadband (spread spectrum) receiver was then investigated and is also documented herein. This work demonstrated that by employing the filterbank to excise (set to zero) those frequency bins contaminated by interference, link performance degradations due to interference could be reduced to the much lesser degradation that results from the loss of desired signal energy in the excised bins. Performance advantages in excess of 20dB were obtained.

Dissemination of this document will raise awareness of the availability and utility of this type of signal processing tool within the Defence community.

## Authors

### **K. L. Harman**

Command, Control, Communications and  
Intelligence Division

*Kevin was awarded a Bachelor's Degree with Honours in Electrical & Electronic Engineering from the University of Adelaide in 1991. He joined DSTO in 1998 as a researcher in the domain of communications signal processing, focussing on wideband digital modem architectures and implementations.*

---

### **C. Potter**

Command, Control, Communications and  
Intelligence Division

*Cindy received her M.E. (RF Engineering), B.Sc. (Maths/Computer), and B.E. (Hons., Electrical & Electronic) from the University of Adelaide in 2002, 1995, and 1994 respectively. As a design engineer with DSTO she is involved in both the RF and digital realms, undertaking design work including the implementation of signal processing algorithms in FPGA, and RF & digital integrated circuit design. Her current area of interest is in the design of circuits in Silicon on Sapphire.*

# Contents

## ABBREVIATIONS AND ACRONYMS

<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>1.1 Scope .....</b>	<b>1</b>
<b>1.2 Overview .....</b>	<b>1</b>
1.2.1 Filterbank Architecture Specification .....	1
1.2.1.1 Weight, Overlap and Add (WOLA) Filter Bank .....	2
1.2.1.2 Analysis Frame Store .....	4
1.2.1.3 Weighting .....	4
1.2.1.4 Coefficient Stores .....	5
1.2.1.5 Inverse WOLA Filter Bank .....	5
1.2.2 Filterbank Core Deliverables .....	5
1.2.3 The DSTO Digital Processor Card.....	6
1.2.4 Integration and Acceptance Testing .....	7
1.2.4.1 Integration .....	7
1.2.4.2 Acceptance Testing.....	8
1.2.5 Excision Filtering Applications.....	9
 <b>2. ACCEPTANCE TESTING.....</b>	 <b>11</b>
<b>2.1 Test Procedure .....</b>	<b>11</b>
<b>2.2 512-bin Filterbank Acceptance Testing .....</b>	<b>13</b>
2.2.1 Test Result – Pseudo-random noise #1 .....	14
2.2.2 Test Results – Other cases .....	16
2.2.3 Confidence testing with the DSSS Demodulator .....	17
2.2.4 Statement of Acceptance.....	18
<b>2.3 4096-bin Filterbank Acceptance Testing .....</b>	<b>18</b>
2.3.1 Test Result – Pseudo-random noise #2 .....	18
2.3.2 Test Results – Other cases .....	20
2.3.3 Confidence testing with the DSSS Demodulator .....	22
2.3.4 Statement of Acceptance.....	22
 <b>3. EXCISION FILTER APPLICATIONS.....</b>	 <b>24</b>
<b>3.1 Scope and Procedure .....</b>	<b>24</b>
<b>3.2 Degradation of the DSSS Demodulator in the presence of Interference.....</b>	<b>25</b>
3.2.1 Theoretical .....	25
3.2.2 Measured .....	27
3.2.2.1 Single tone jammer versus band position and relative power.....	27
3.2.2.2 Multi-tone jammer versus relative power .....	27
<b>3.3 Excision Filtering with the 4096-bin Filterbank.....</b>	<b>30</b>
3.3.1 Single-tone Excision .....	30
3.3.2 Multi-tone Excision .....	30
<b>3.4 DSSS Demodulation Degradation in the presence of Bin-nulling.....</b>	<b>33</b>
<b>3.5 Excision Filtering with the 512-bin Filterbank.....</b>	<b>34</b>
<b>3.6 Conclusions.....</b>	<b>34</b>

<b>4. REFERENCES.....</b>	<b>36</b>
<b>APPENDIX A: 512-BIN FILTERBANK ACCEPTANCE TEST RESULTS .....</b>	<b>37</b>
A.1. 100 kHz Tone.....	37
A.2. 500 kHz Tone.....	39
A.3. 597.6 kHz Tone.....	40
A.4. 402.4 kHz Tone.....	42
A.5. 1.000 MHz Tone .....	43
A.6. 5.000 MHz Tone .....	45
A.7. 10.000 MHz Tone .....	46
A.8. 25.000 MHz Tone .....	48
A.9. 40.000 MHz Tone .....	49
A.10. 50.000 MHz Tone .....	51
A.11. Multi-tone Case#001.....	52
A.12. Multi-tone Case#002.....	54
A.13. Multi-tone Case#003.....	55
A.14. Multi-tone Case#004.....	57
A.15. Stored Waveform – Square Wave.....	58
A.16. Stored Waveform – Ramp.....	60
A.17. AWGN, 200MSPS.....	61
<b>APPENDIX B: 4096-BIN FILTERBANK ACCEPTANCE TEST RESULTS .....</b>	<b>63</b>
B.1. 200 kHz Tone.....	63
B.2. 206 kHz Tone.....	65
B.3. 212 kHz Tone.....	66
B.4. 218 kHz Tone.....	68
B.5. 224 kHz Tone.....	69
B.6. 1 MHz Tone .....	71
B.7. 2 MHz Tone .....	72
B.8. 5 MHz Tone .....	74
B.9. 10 MHz Tone .....	75
B.10. 20 MHz Tone .....	77
B.11. 22.857373 MHz Tone .....	78
B.12. 50 MHz Tone .....	80
B.13. Multi-tone Case#001.....	81
B.14. Multi-tone Case#002.....	83
B.15. Multi-tone Case#003.....	84
B.16. Multi-tone Case#004.....	86
B.17. Stored Waveform – Square Wave.....	87
B.18. Stored Waveform – Ramp.....	89
<b>APPENDIX C: MATLAB SCRIPTS.....</b>	<b>91</b>
C.1. Error Vector Magnitude Measurement .....	91
C.2. Test Result Log and Analysis.....	95

## Abbreviations and Acronyms

<b>ADC</b>	Analogue-to-digital converter
<b>AGC</b>	Automatic gain control
<b>AWGN</b>	Additive white Gaussian noise
<b>BER</b>	Bit error rate
<b>bps</b>	Bits per second, used with SI prefixes
<b>dB</b>	Decibel
<b>dBm</b>	Decibels relative to 1 milliwatt
<b>DC</b>	Direct current
<b>DCC</b>	Data conversion and clocking
<b>DDP</b>	Digital data processor
<b>DDS</b>	Direct digital synthesis/synthesiser
<b>DFT</b>	Discrete Fourier transform
<b>DPC</b>	Digital processor card
<b>DSP</b>	Digital signal processing
<b>DSSS</b>	Direct sequence spread spectrum
<b>DSTO</b>	Defence Science and Technology Organisation
<b>Eb/No</b>	Ratio of energy per bit (or chip) to noise power spectral density
<b>EVM</b>	Error-vector magnitude
<b>FPGA</b>	Field-programmable gate array
<b>GUI</b>	Graphical user interface
<b>Hz</b>	Hertz (cycle/second), used with SI prefixes
<b>IP</b>	Intellectual property
<b>IWOLA</b>	Inverse WOLA
<b>JTAG</b>	Joint Test Action Group Interface standard
<b>LE</b>	Logic element
<b>LVC MOS</b>	Low-voltage CMOS
<b>LVDS</b>	Low-voltage differential signalling
<b>MCPS</b>	Mega-chips per second
<b>PCI</b>	Peripheral component interconnect bus standard
<b>pp</b>	Peak-to-peak
<b>ppm</b>	Parts per million
<b>PRBS</b>	Pseudo-random binary sequence
<b>RAM</b>	Random access memory
<b>RF</b>	Radio frequency
<b>RFEL</b>	RF Engines Limited
<b>Rx</b>	Receiver
<b>s</b>	Second, used with SI prefixes
<b>SPS</b>	Samples per second, used with SI prefixes
<b>SW</b>	Software
<b>.tff</b>	Tabular text file format
<b>Tx</b>	Transmitter

<b>V</b>	Volts, used with SI prefixes
<b>vs</b>	versus
<b>VSG</b>	Vector signal generator
<b>WOLA</b>	Weighted overlap and add
<b><math>\sigma</math></b>	Standard deviation
<b><math>\Omega</math></b>	Ohms



# 1. Introduction

## 1.1 Scope

In 2005 DSTO contracted “RF Engines Limited” (RFEL), an intellectual property (IP) core design company with renowned experience in implementing field-programmable gate array (FPGA) based discrete Fourier transforms (DFT) and related solutions, to design and deliver two customised filterbank cores.

RFEL [1] describes “the basic function of (these cores) is to perform high performance adaptive filtering on a complex input signal, by first converting it to a frequency domain representation, then weighting the signal with a set of user programmed coefficients, and finally converting the signal back to the time-domain.”

DSTO will develop such algorithms in support of Defence client wireless communications tasks by applying these cores on an existing custom-designed FPGA-based signal processing platform designated the Digital Processor Card (DPC).

Prerequisite to algorithm development and application are the integration of the cores into a DPC-based system and subsequent validation of the cores with respect to the contractual performance requirements (acceptance testing).

Application of the filterbank was then initially envisaged in conjunction with a wideband direct-sequence spread-spectrum (DSSS) demodulator core developed by DSTO for the DPC [2]. Excision filtering by bin-nulling is an implementation-trivial example with significant performance potential.

In this document the basic architecture of the filterbank is reviewed, and the processes and results of its integration, acceptance testing, and application as an excision filter are described.

## 1.2 Overview

### 1.2.1 Filterbank Architecture Specification

Details of the contract with RFEL are given in [3].

A thorough discussion of filterbank architectures and performance is given in [4, 5].

In this section, extracts from [1] and [6] are reproduced for completeness.

The main processing blocks and interfaces of the filterbank core are shown in Figure 1. This block diagram, while representative of the signal processing functionality, is not necessarily representative of the actual hardware implementation. Each of the blocks is described in the following sub-sections.

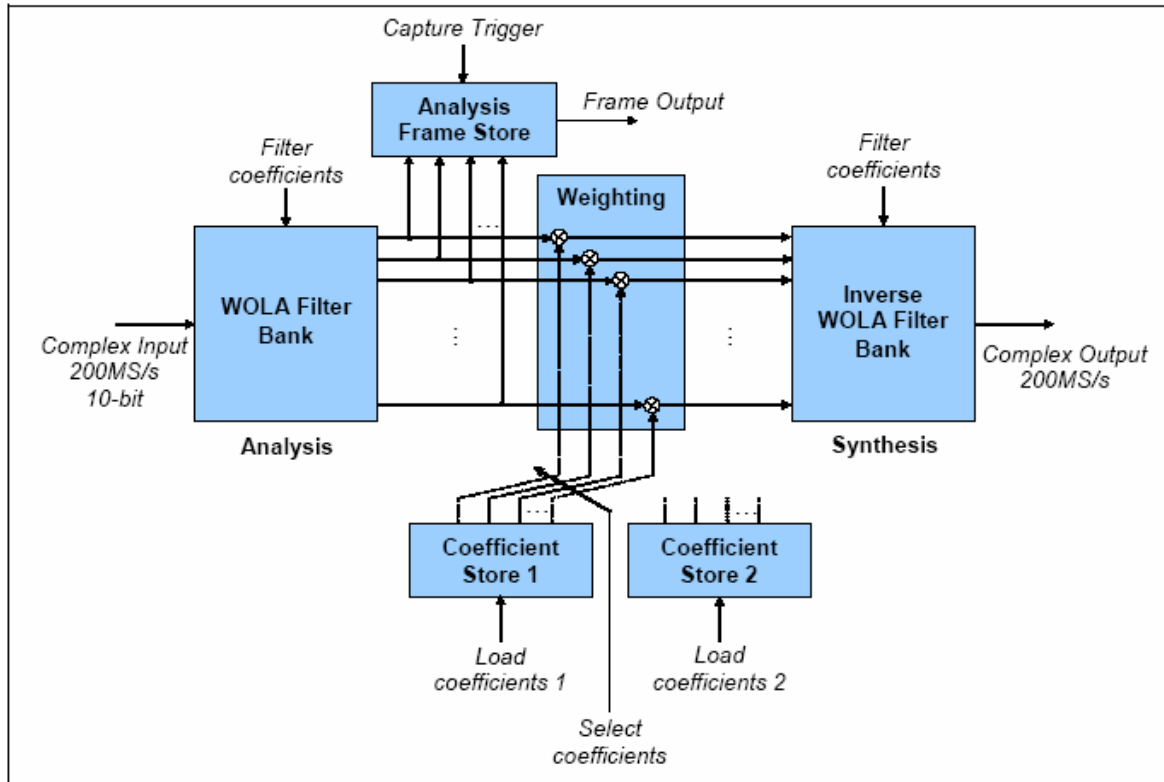


Figure 1- RFEL filterbank IP functions and interfaces

#### 1.2.1.1 Weight, Overlap and Add (WOLA) Filter Bank

The WOLA (Weighted Overlap and Add) filter is an efficient method used to implement a uniformly distributed multi-channel filter bank, such as the Polyphase DFT. The architecture consists of a polyphase front-end that performs the weighting, overlapping and addition followed by a DFT that is usually implemented using an FFT. The WOLA is generally used in applications that demand high quality filters in terms of stop band rejection and filter shape.

Figure 2 shows a WOLA architecture that can be used to implement the Polyphase DFT, and is representative of the hardware implementation. The incoming complex samples are continuously shifted into an input register of length  $L$ . After a period of  $M$  input samples, the  $L$  stored samples are weighted by the  $L$  prototype coefficients  $h(n)$ . These weighted samples are then split into  $L/K$  blocks of  $K$  samples, and added sample-wise to form the block of  $K$  input samples to the DFT. The DFT part of the Polyphase DFT is implemented using pipelined FFT Cores. The ratio of  $K/M$  determines the output sample rate of the filter bank ( $K=M$  for critical sampling,  $K=2M$  for twice over-sampling etc.)

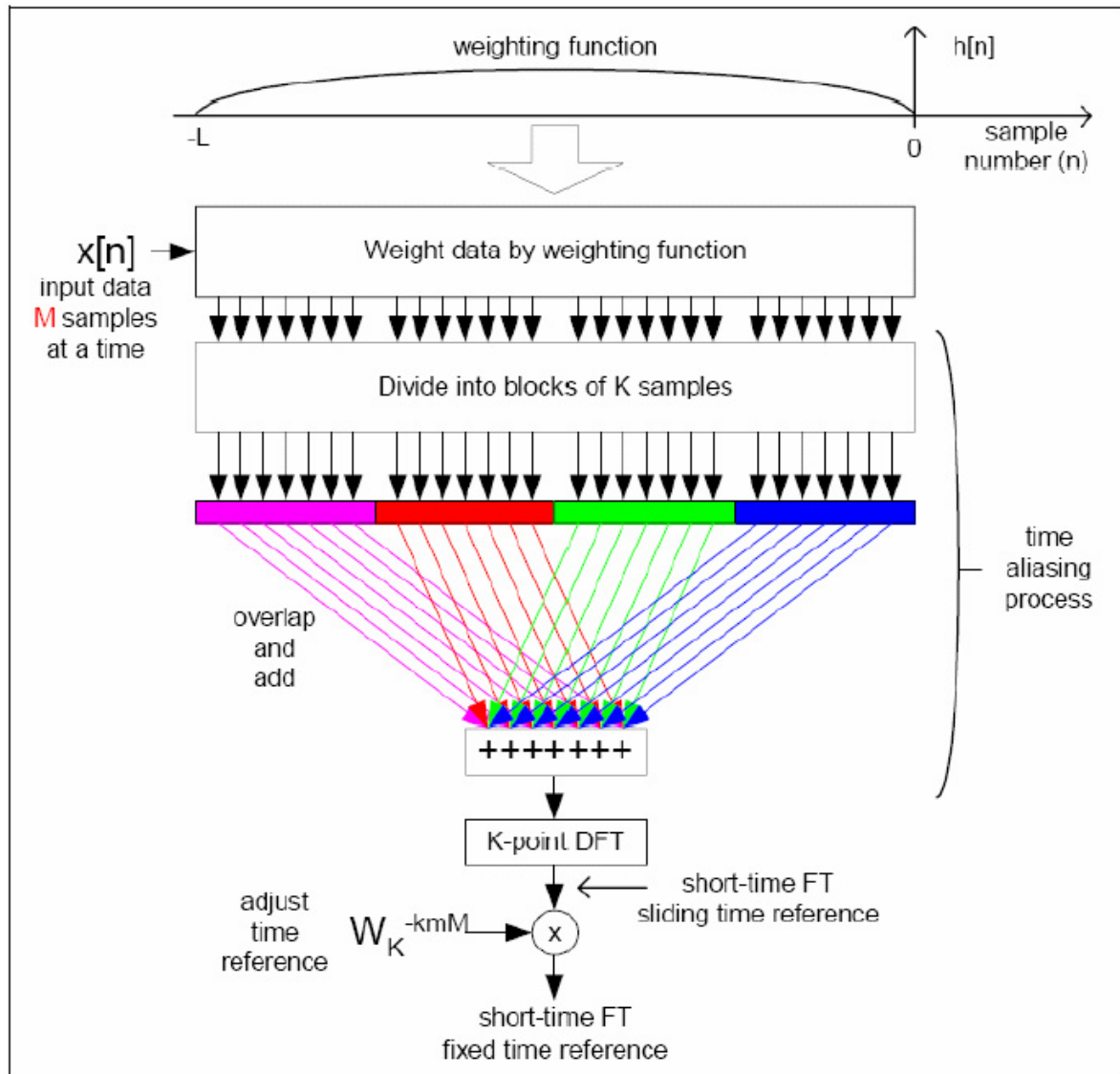


Figure 2- WOLA architecture

The following constraints apply to the contracted WOLA implementation:

- The overlap between channels will be constrained such that leakage into non-adjacent channels is less than a desired threshold  $M_{LEAKAGE}$ .
- The sum of the squares of each channel will closely approximate the power of the input signal.
- Back-to-back configuration of the analysis and synthesis sections will yield a reconstructed time-domain signal with error vector magnitude (EVM) better than a desired limit  $M_{EVM}$  error with respect to the input signal.
- Filter coefficients are to be loaded after the core is loaded.
- The output from the WOLA will be  $K$  channels.
- The filter length will be  $4K$ .

Prototype coefficients  $h(n)$  have been designed by DSTO [12] which will yield the filter response shown in Figure 3.

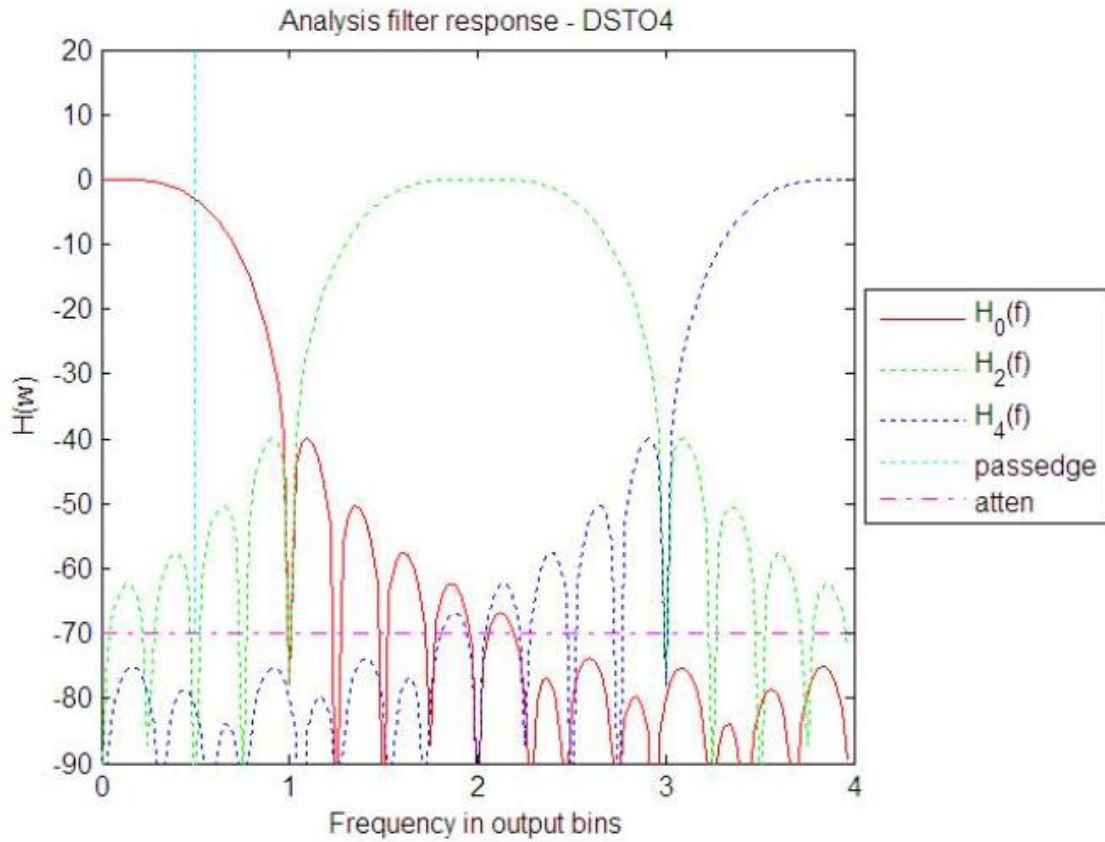


Figure 3- DSTO analysis and synthesis filter responses

#### 1.2.1.2 Analysis Frame Store

The output of the WOLA filter bank can be latched into a buffer of length  $K$  complex samples under user control. The purpose of this buffer is to allow the user to capture a frame of output data from the WOLA (an analysis frame), which may then be analysed in a separate part of the system. The user will tell the core to capture a new frame of data using the trigger input, and then read the contents of the buffer at some time later. The output of the store will be in natural frequency order. It is not intended that the capture rate allow contiguous analysis frame capture.

#### 1.2.1.3 Weighting

The complex channel outputs from the WOLA are to be independently weighted by a set of  $K$  coefficients, in order to perform frequency domain filtering.

### 1.2.1.4 Coefficient Stores

A dual buffer approach will be employed for storing coefficients, which will enable coefficients to be updated at run-time without interruption of the core processing. Whilst the coefficients are being used from one buffer, the user will be free to load new coefficients into the other. Switching to the new set of coefficients will occur under user control, but the core will wait until the current frame has finished before switching over to the new coefficients.

### 1.2.1.5 Inverse WOLA Filter Bank

The Inverse WOLA (IWOLA) performs the opposite function to the WOLA filter bank. It takes the  $K$  channels, and re-synthesises these to form a single time-domain sequence, with a sample rate of  $F_s$  (complex). WOLA and IWOLA filters will be designed in close conjunction in order to achieve the  $M_{EVM}$  limit.

## 1.2.2 Filterbank Core Deliverables

Two cores were delivered and are designated the 512-bin and 4096-bin versions.

The 512-bin version has the following specifications:

- $K = 512$
- $F_s \leq 200\text{MSPS}$  (complex)
- $M_{\text{LEAKAGE}} \leq -70\text{dB}$
- $M_{\text{EVM}} \leq -40\text{dB}$ .

It should process and manipulate a 200 MHz (null-to-null) signal bandwidth without aliasing, represented with 512 bins, each of width  $W_{512} = 200\text{e}6/512 \approx 390.6$  kHz. The bins may be interpreted as spanning the frequency range -100.000 MHz (Bin#1) to +99.610 MHz (Bin#512), with the DC component at Bin#257.

The 4096-bin version has the following specifications:

- $K = 4096$
- $F_s \leq 100\text{MSPS}$  (complex)
- $M_{\text{LEAKAGE}} \leq -70\text{dB}$
- $M_{\text{EVM}} \leq -40\text{dB}$ .

It should process and manipulate a 100 MHz (null-to-null) signal bandwidth without aliasing, represented with 4096 bins, each of width  $W_{4096} = 100\text{e}6/4096 \approx 24.4$  kHz. The bins may be interpreted as spanning the frequency range -50.000 MHz (Bin#1) to +49.976 MHz (Bin#4096), with the DC component at Bin#2049.

An interface specification for the cores is available [7].

### 1.2.3 The DSTO Digital Processor Card

The DPC is described in [8] as a highly configurable and computationally powerful signal processing engine. It consists of a mainboard and up to four interchangeable modules.

The mainboard is a universal (3.3v or 5v) PCI plug-in card with extended length form-factor that supports 64/32-bit and 66/33- MHz modes of operation and is compliant with revision 2.2 of the PCI standard. The mainboard features include:

- A PCI Interface Target, implemented as a custom firmware (FW) entity in an Altera EP1S10F484C5 FPGA.
- Four module sites designated Slot#0 to Slot#3, each of which can host a plug-in module. A pair of high-density connectors for each slot facilitates Target-to-Slot, Slot-to-Slot (adjacent Slots only) and power connectivity.
- Back-end (application-side) PCI signalling between the Target and the Slots, allowing PCI access to each plug-in module.
- Custom signalling between the Target and the Slots, intended primarily as an FPGA configuration bus when plug-in modules contain volatile FPGAs.
- JTAG interfacing to all FPGAs.
- Local power generation and distribution, with the PCI power rails as inputs.
- A PCI back-panel with analogue I and Q inputs (SMA F/M), a reference frequency output (SMA F/M), and a JTAG port (12-pin header, M).

There are two plug-in modules pertinent to filterbank applications, namely a Data Conversion and Clocking (DCC) module and a Digital Data Processing (DDP) module.

The DCC is essentially an analogue-to-digital converter (ADC) module, and also provides the primary signal processing clock. It features:

- High-stability clock generation and distribution. It provides a fixed-rate, 200 MHz sample clock with 100ppm stability and 1ps of 1- $\sigma$  phase jitter.
- Two channels of synchronous analogue to digital conversion with 10-bit, 200MSPS ADC devices identified as the in-phase (I) and quadrature-phase (Q) channels (consistent with application in quadrature modulation schemes). The ADC output on each channel is presented in a demultiplexed form at half rate, together with a sample-synchronous 100 MHz output clock. The demultiplexed channels are designated as channel A and channel B, leading to four 100MSPS output streams designated IA, IB, QA and QB, each with 10-bit resolution.
- External (adjacent Slot) control of the ADC enable, output data format and output data interleaving controls.
- Local power generation and conditioning.

The DDP is the main signal processing engine, hosting a high-density FPGA for FW-based operations. It features:

- An Altera EP1S80F1508C6 FPGA. This provides approximately 80,000 LE, 7,427kbit RAM, and 22 embedded DSP Blocks for up to  $176 \times 9$ -bit multipliers. It supports clock rates in excess of 200 MHz.
- Connection to the mainboard back-end PCI and configuration buses.
- High-density signalling to adjacent Slots.
- A 42-way LVDS (or 84-way LVCMOS) Digital Input/Output (I/O) Header for general purpose I/O.
- Local power generation and conditioning.
- Test headers.

When hosting the filterbank, the DPC would normally be configured with a DCC in Slot#0, a DDP in Slot#1 with the integrated filterbank core, and a DDP in Slot#2 with another application such as a FW demodulator core which accepts sampled inputs that may have been pre-processed by the filterbank.

## 1.2.4 Integration and Acceptance Testing

### 1.2.4.1 Integration

Each of the 512-bin and 4096-bin *cores* are integrated into an overall FW *application* compatible with a Slot#1 DDP.

The 512-bin filterbank application is shown in Figure 4, where:

- The stream processing nature of the application is apparent. Sampled digital data and the sample clock reference are connected at the input, possibly modified by the filterbank, and then reconnected at the output, being essentially transparent to subsequent signal processors in the chain.
- The “Clock Generation & Distribution” function facilitates both decimation of data rate, if desired, and optimal clock phase alignment internal to the FPGA.
- An inbuilt “Waveform Function Generator” facilitates built-in-test of the filterbank core. On each of the I and Q channels it can independently generate: all zeroes; all ones; a square wave; a ramp; or a pseudo-random noise function.
- The “Data Capture RAM” facilitates monitoring of blocks of sampled data directly at the input or output of the filterbank, and can also be used to retrieve the analysis frame snapshot for external processing. Samples within a block are time-contiguous, but samples between blocks are not so. The maximum capture depth is 32768 (complex) samples.
- Control and monitoring of these functions is realised via the PCI interface to the Slot, from which a set of independently addressable “Status & Control Registers” is accessible at the PCI transaction rate.
- There is a dedicated “Filterbank Control Interface” for control and monitoring transactions where the handshaking protocol is too time critical for register-only access.

In this application, the IP core alone requires 46% of the DDPs logic capacity, 82% of its DSP blocks and 11% of its RAM.

The instantiation of the 4096-bin IP core is similar, except its resource requirements impinge on the embedded memory available for the “Waveform Function Generator” and “Data Capture RAM”. The core alone requires 56% of the DDPs logic capacity, 91% of its DSP blocks and 70% of its RAM. Consequently the pseudo-random noise function is not available and the maximum capture depth is reduced to 8192 (complex) samples.

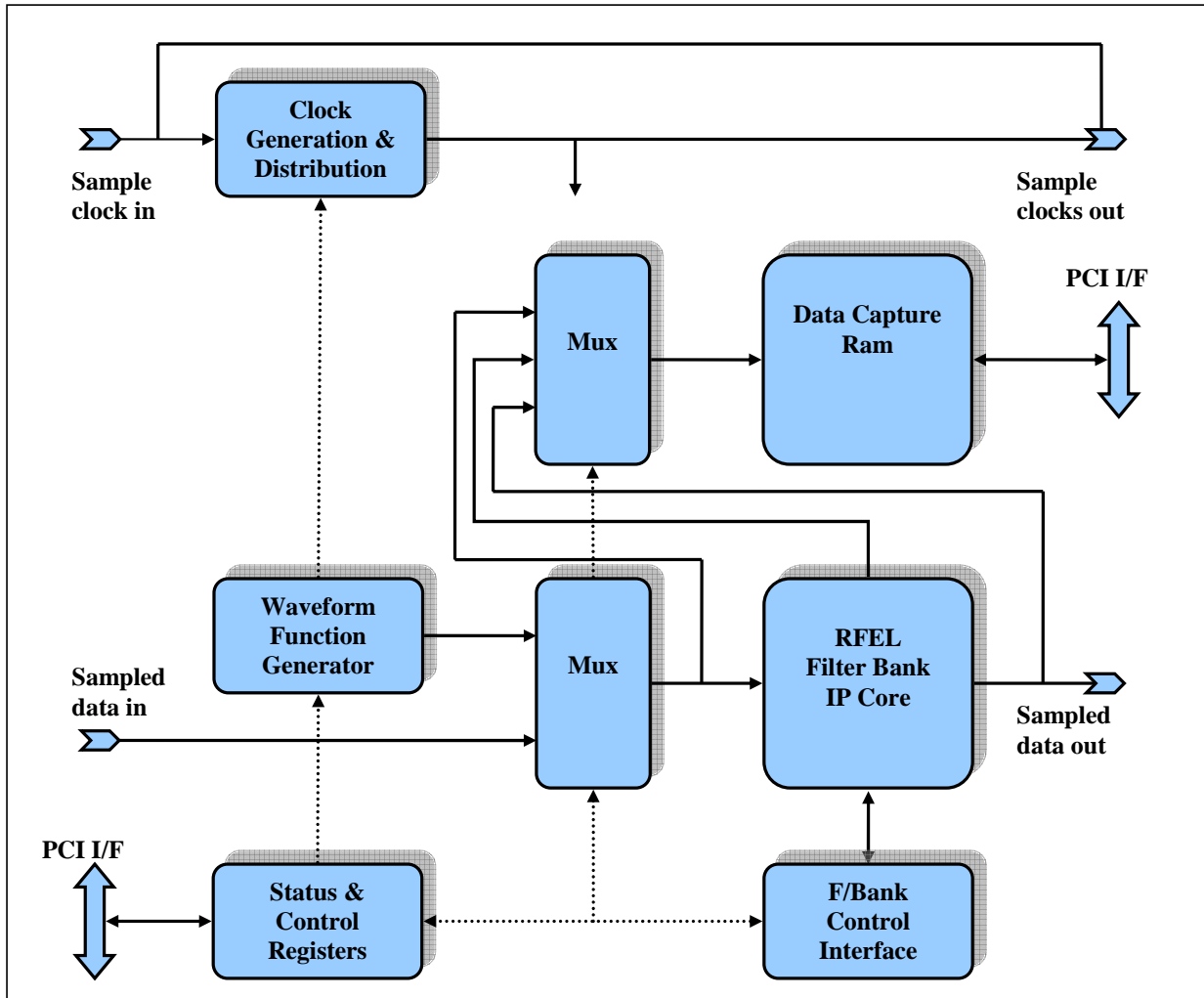


Figure 4 – FW components of the 512-bin filterbank application.

#### 1.2.4.2 Acceptance Testing

The primary figure of merit for acceptance testing is the per sample output signal reconstruction error with respect to the input signal, denoted the “error vector magnitude” or EVM. The cores were designed for reconstruction with  $\text{EVM} \leq -40\text{dB}$ .



To test the EVM performance, a block of output signal samples should be captured in conjunction with the input signal samples that originated those outputs, allowing for processing latencies through the core. The magnitude errors can then be compared on a sample-by-sample basis using a tool such as MATLAB [11]. In the applications described in the preceding section, either the input or output of the filterbank core may be captured, but not both simultaneously.

Accordingly an additional application designated “Slot#2 Data Capture” was developed for Slot#2 of the DPC. The filterbank application is configured to log the input data to the filterbank core, while the Slot#2 Data Capture application logs the output data from the core, passed without modification from Slot#1 to Slot#2 on DPC buses. In architecture, the Slot#2 Data Capture core is similar to that of the filterbank applications, except that the “Waveform Function Generator” and filterbank core are not required.

Synchronisation between the filterbank and data capture circuits is achieved by inclusion of an additional real-time control signal, routed on the DPC buses between Slot#1 and Slot#2, which is a duplicate of the capture trigger signal used internally by the filterbank application when it commences data logging. For the 512-bin case, where the maximum capture depth is well in excess of the latency, no other synchronisation is required, and the input and output sample files can be aligned by correlation testing. For the 4096-bin case, an additional programmable delay on the trigger signal at Slot#2 was added and tuned to match the latency, thereby eliminating unusable uncorrelated file segments.

In addition to the signal capture tools, a suite of test signals is required which will ensure good coverage of the cores operating range. The following signal types were chosen:

- *Full-scale sinusoid*. This allows testing of tone reproduction at any desired frequency in the band, including at fractional bin positions.
- *Multi-tone sinusoid*. Validates more complicated signal reproduction. These signals were produced using an arbitrary waveform generator from files generated in MATLAB as a weighted sum of sinusoids.
- *Full-scale square wave*. Uses the internal function generator.
- *Full-scale ramp*. Uses the internal function generator.
- *Pseudo-random wideband noise – internal*. Uses the internal function generator. Tests reproduction with simultaneous spectral occupancy at all band positions.
- *Pseudo-random wideband noise – external*. Uses a commercial off-the-shelf (COTS) noise generator for a different noise distribution.

The EVM results from applying these signal types to both filterbanks are given in Section 2.

### 1.2.5 Excision Filtering Applications

A useful application for the filterbank is the excision filtering of interfering narrowband signals in a spectrum of interest. By observation of the analysis frame store and with knowledge of the intended or desired spectrum, the locations of interferers can be identified and the complex weighting for afflicted bins can be set to zero to eliminate those

signals in the reconstructed output. This approach is simple but has the penalty that the energy of the desired signal is also lost from those bins. More sophisticated weighting algorithms, which better approximate the Wiener filter by retaining a proportion of that bin energy consistent with some estimate of the desired signals spectrum, are possible [12] but are not considered in this report.

Narrowband excision complements the DSTO DSSS demodulator, which uses a broadband modulation that is highly likely to have in-band interferers. The DSSS demodulator exists as a DPC-ready firmware core and accompanying graphical user interface (GUI) software, and hence is conveniently testable on the same platform as that used for filterbank acceptance testing.

In this report, an investigation into filterbank excision filtering with the DSSS receiver is conducted with emphasis on:

- The expected degradation of a DSSS demodulation in the presence of narrowband interference, based on established theory.
- Test of the DSSS receiver in the presence of controlled jammers, again as a function of band position and relative power.
- Test of the utility of filterbank excision in the presence of the same jammers.
- Limitations of the approach due to receiver AGC (automatic gain control).
- Limitations of the approach due to intentional signal power loss by nulling.

## 2. Acceptance Testing

### 2.1 Test Procedure

Acceptance testing aimed to stimulate the filterbank IP cores with known input samples and verify that these input samples are reconstructed at the filterbank output with error magnitude no greater than -40dB.

The test arrangement is shown in Figure 5. The core under test was integrated in the filterbank application FW, which was hosted on a DPC. The DPC was hosted in a standard desktop computer, running graphical user interface (GUI) software which allows control and monitoring of DPC applications. Test signal generation was performed with either a function generator external to the computer, or using the waveform generator internal to the filterbank application. When using the external sources, signals were connected to the DCC module in Slot#0 of the DPC, where they were digitised and routed across to the filterbank in Slot#1. Optional attenuation in the path controlled clipping at the ADC stage. When using internal sources, the digital samples were generated directly in Slot#1. By a combination of local RAM storage in Slot#1 and PCI transactions with the host, sampled data directly at the filterbank IP core input could be logged for post processing. Core outputs were routed directly from Slot#1 to a Data Capture application in Slot#2, where again a combination of local RAM storage and PCI transactions allowed filterbank IP core output to be logged for post processing.

Using this, the test procedure was as follows:

- Configure the signal source (and attenuation, if required).
- Using the GUI, specify a log file name in which the input and output samples will be logged.
- Using the GUI, specify the log file depth (number of complex samples).
- Using the GUI, specify the core latency.
- Using the GUI, trigger a capture event.
  - *The filterbank and Data Capture RAM are prepared for writing.*
  - *A trigger event is sent to Slot#1. This triggers both real-time logging of the input samples to RAM in Slot#1 and a real-time secondary trigger to Slot#2.*
  - *Slot#2 receives the secondary trigger and initiates its latency counter.*
  - *Upon expiry of the latency counter, real-time logging of the output samples to RAM commences.*
  - *When the required number of samples completes in both slots, the RAMs are read in turn to the log file.*
- Using standard tools, transfer the log files for post-processing.

Post processing was implemented with MATLAB in accordance with the following algorithm:

- Open the log file for reading.
- Read the log file data sets into named vectors.

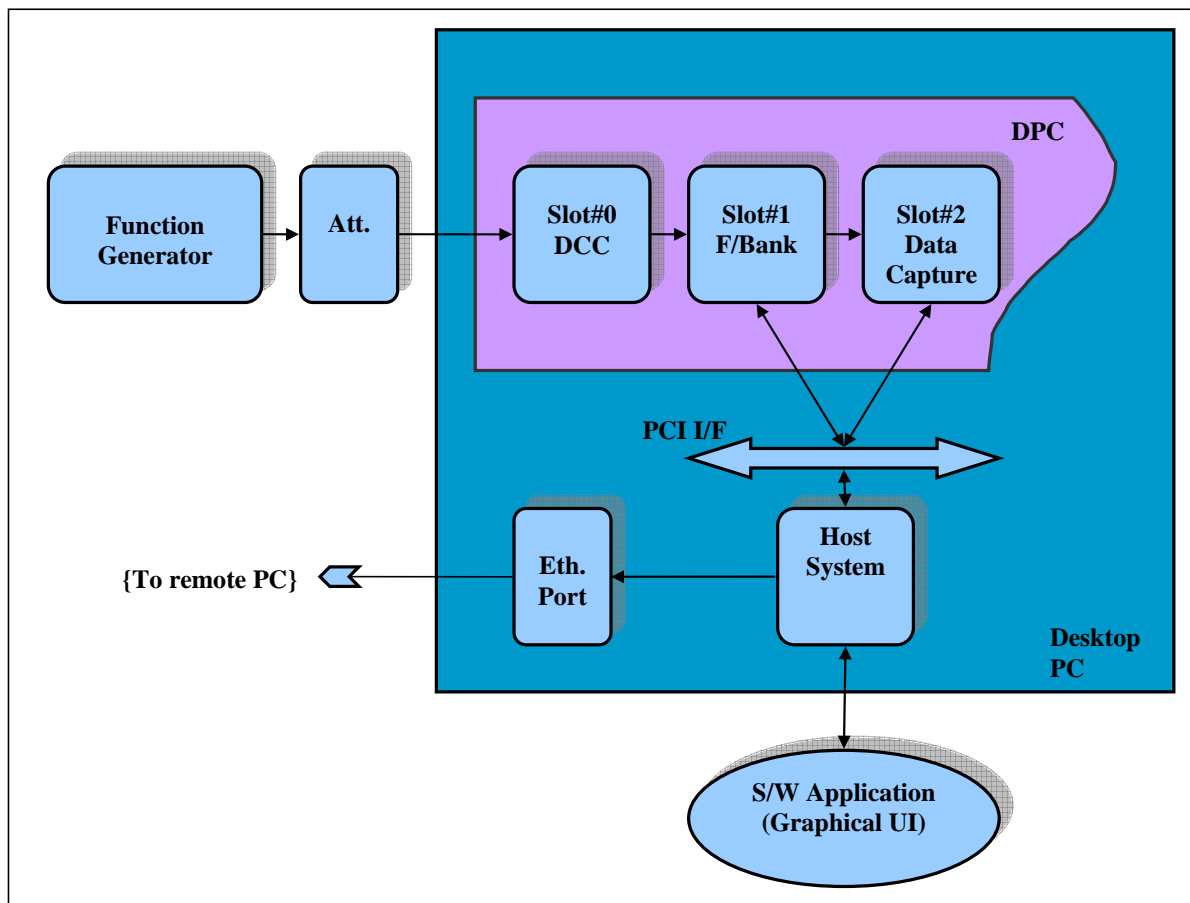


Figure 5 – Test Arrangement for the EVM ATP

- Truncate the beginning of the input vectors in accordance with the latency.
- Truncate the end of the output vectors in accordance with the latency.
- Normalise each vector with respect to its absolute maximum value.
- Scale the output vectors in accordance with the ratio of the standard deviations of the input and output vectors.
- Compute the EVM as  $20 \cdot \log_{10}(\text{normalised output} - \text{normalised input})$ .

The input sample sets should provide high coverage of the operating range of the cores. The real and imaginary inputs were driven by independent generators with the same signal types. Signal types used are described in Table 1.

Table 1 – ATP Test Signals

Type	Source	Scope
Tone	Agilent 33250A Arbitrary Waveform Generator, Sine mode, 130mVpp into 50Ω.	Test simple signal reconstruction.  Vary frequency to test any bin or fractional bin position.
Multi-tone	Rohde & Schwarz SMU200A VSG, ARB mode, rear-panel I/Q out with 10dB pads.	Test incrementally more complicated signal reconstruction.
Square wave	Internal waveform, Scaled -511...+511.	Test broadband signal reconstruction, known perfect full-scale input.
Ramp	Internal waveform, Scaled -511...+511.	Test broadband signal reconstruction, known perfect full-scale input.
Pseudo-random noise #1	Internal waveform.	Test broadband signal reconstruction with full bandwidth occupancy.
Pseudo-random noise #2	Rohde & Schwarz SMU200A VSG, AWGN mode, rear-panel I/Q out with 10dB pads.	Test broadband signal reconstruction with full bandwidth occupancy. This second scenario provides a different and potentially more random stimulus than the precomputed waveform in case #1.

## 2.2 512-bin Filterbank Acceptance Testing

The firmware applications used in this test procedure have the following designations:

- Filterbank: *rfel\_filterbank\_ver3\_190706\_02.ttf*
- Data Capture: *Slot2\_dataCapture\_201106\_02.ttf*

Although the 512-bin core is rated for operation at 200MSPS, the majority of testing was conducted at 100MSPS to alleviate thermal stresses on the DPC. The core design is wholly synchronous and therefore rate scalable, hence testing at the lower sample rate is equally valid. Select testing was performed at full rate, but only to verify the FPGA place and route timing performance.

When performing signal reconstruction tests, the filterbank's analysis frame weights were set to unity, affecting a transparent mode of operation.

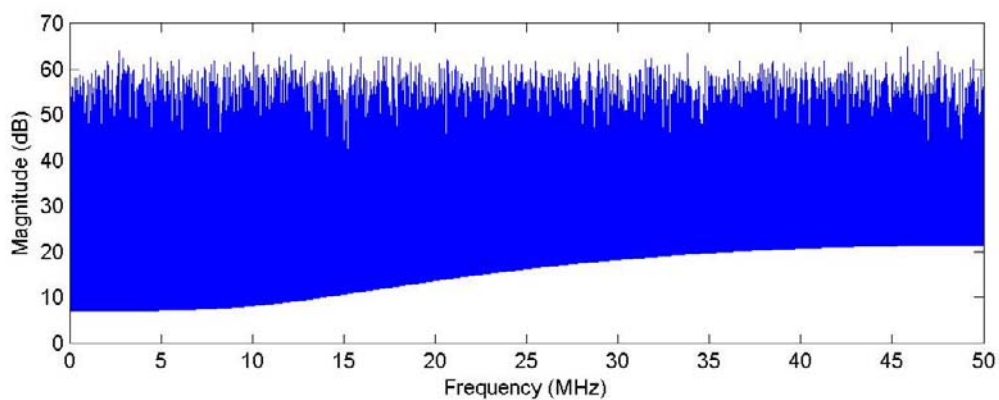
### 2.2.1 Test Result – Pseudo-random noise #1

A graphical interpretation of the measured signals is given in **Error! Reference source not found.**. The worst-case EVM is -47.0dB. This is a significant result as it confirms the filterbank core's capability to reconstruct a complicated signal with components in all bins and of varying magnitudes.

Table 2 – Tests result visualisation – 512-bin core with pseudo-random noise I/O

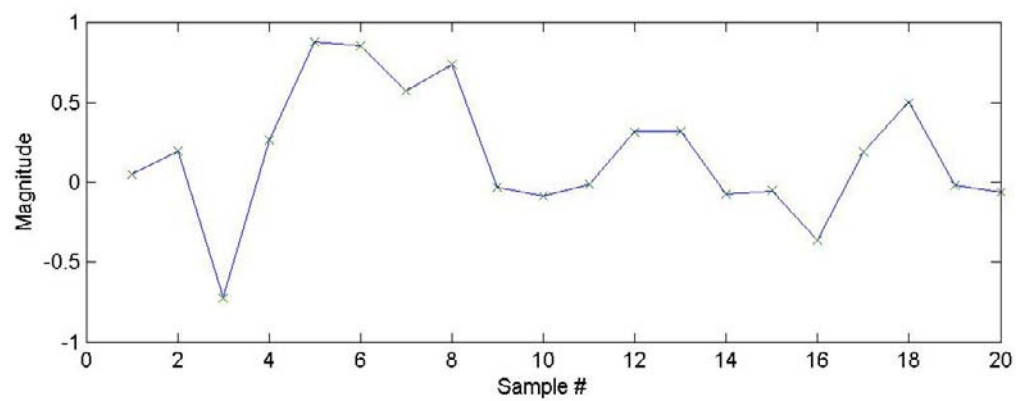
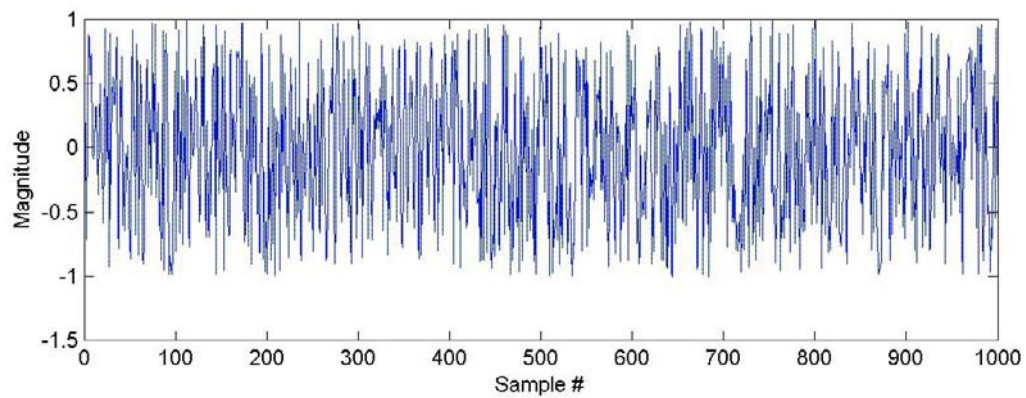
**Description:** Test of broadband signal reconstruction, stored pseudo-random noise.

**Input signal power spectrum:**

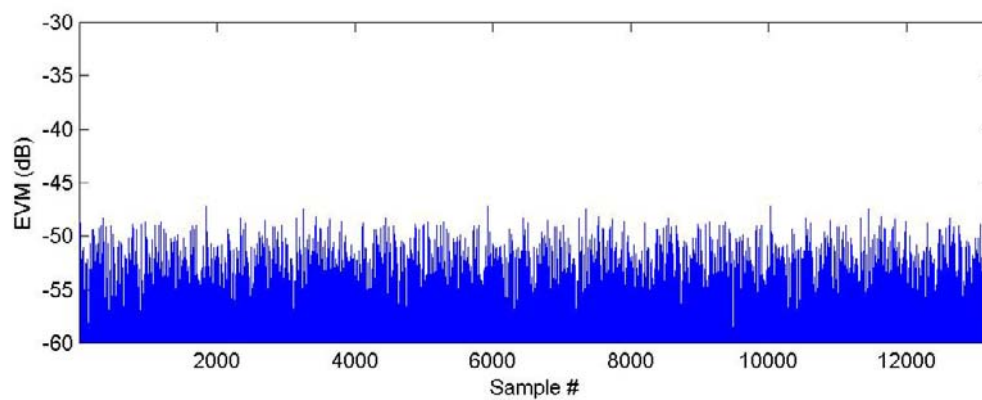


**Description:** Test of broadband signal reconstruction, stored pseudo-random noise.

**Typical time domain input vs output:**



**Typical EVM:**



**Peak EVM:** -47.0dB

### 2.2.2 Test Results – Other cases

For improved readability, other graphical results are segregated to Appendix A: .

A summary of the EVM results is given in Table 3.

*Table 3 – Summary of EVM results (512-bin)*

<b>Stimulus</b>	<b>Comment</b>	<b>Peak EVM</b>
100 kHz tone	Bin edge.	-48.1
500 kHz tone	Bin edge.	-47.7
597.6 kHz tone	+1/2 bin (bin-centre)	-46.2
402.4 kHz tone	-1/2 bin (bin-centre)  Tone reproduction quality is independent of fractional bin position.	-47.3
1 MHz tone		-45.5
5 MHz tone	EVM evidences some regularity at 5 MHz and above.	-45.5
10 MHz tone		-44.2
25 MHz tone		-44.7
40 MHz tone		-44.9
50 MHz tone	Some errors in violation of specifications, in the presence of large signals near the aliasing frequency.	-34.6
Multi-tone, Case #1		-45.3
Multi-tone, Case #2		-42.2
Multi-tone, Case #3		-39.7
Multi-tone, Case #4		-46.0
Stored Square Wave	EVM evidences cyclic regularity.	-52.3
Stored Ramp	EVM evidences cyclic regularity.	-52.3
AWGN, 200MSPS	Valid operation at full sample rate and with a complicated signal.	-43.2



### 2.2.3 Confidence testing with the DSSS Demodulator

The filterbank is intended to operate 'transparently', or in-line, with other signal processing streams. An additional measure of confidence in the filterbank's continuous real-time reconstruction performance could be obtained by operating the filterbank in-line ahead of a known good processing function and checking for degradations.

It was proposed to test the cores in-line with the DSTO DSSS demodulator [2], in which case the performance measure is the bit error rate (BER) at any given operating point before and after insertion of the filterbank.

The test arrangement is similar to that shown in Figure 5, except that the external function generator is replaced by a custom DSSS transmitter and RF downconverter, leading to a complex baseband DSSS input signal to the DPC. Also, the Data Capture application in Slot#2 is replaced by a DSSS Demodulator application, and the demodulated bit stream from this slot is assessed externally in real-time with a bit error rate tester (BERT).

The BER without the filterbank is known [2, 9]. The BER with the filterbank is remeasured and shown relative to the original curve in Figure 6, for the case of 2Mbps data rate and 50MCPS chip rate, and using the demodulator version *dssss\_rake\_top\_11.ttf*.

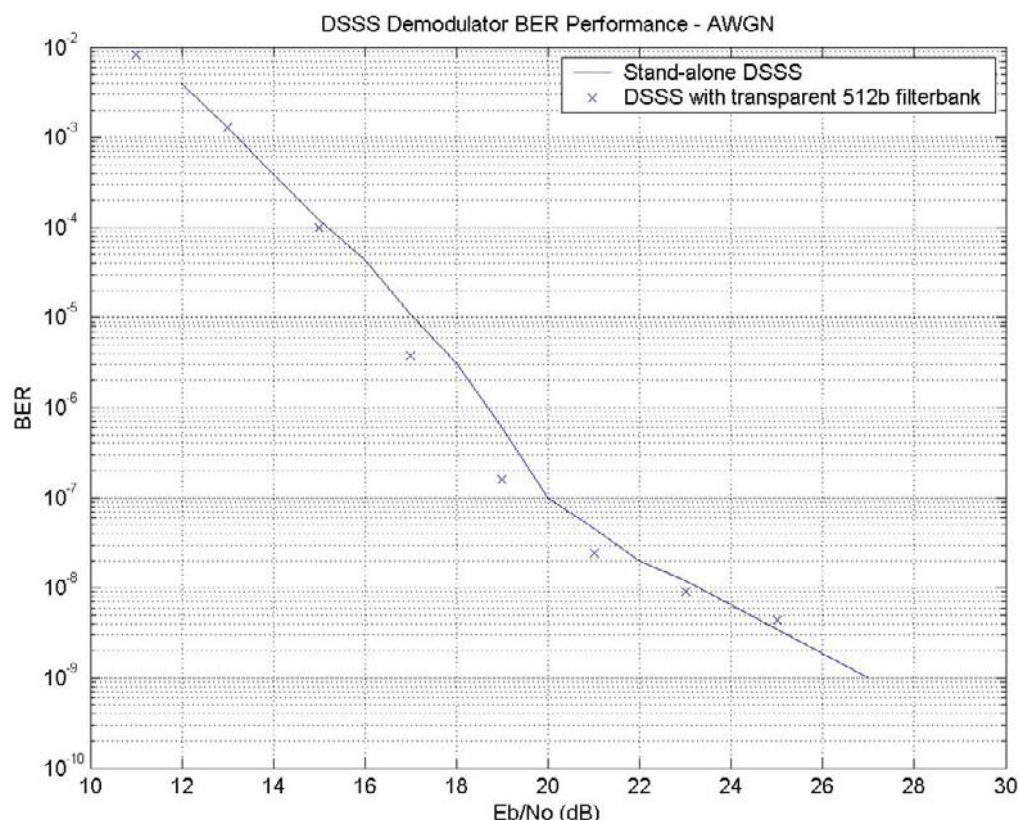


Figure 6 – BER performance of the DSSS demodulator with in-line 512-bin filterbank

These results indicate that there is no appreciable degradation in the demodulator performance when operated in-line with the filterbank, and hence the filterbank's continuous real-time reconstruction performance is validated.

#### 2.2.4 Statement of Acceptance

In general the reconstruction performance of the cores is satisfactory.

However, as suggested by the results of the 50 MHz tone test and Multi-tone (case #3) test, the core performance can be marginal under some conditions. Specifically, near full scale spectral content in the high frequency bins, when operating with full-scale coefficients for the analysis filter, synthesis filter and weights, can cause larger reconstruction errors. This issue was discussed with RFEL, leading to the revelation that the filterbank core's management of numeric overflow conditions is imperfect – the cores are designed to manage a wide variety of input signal conditions correctly, but can be tripped into overflow under some conditions.

It was agreed with RFEL that such events could be managed by reducing either or both of the filter and weight coefficients.

Given these operating constraints (and noting that the as yet untested ability of the cores to correctly manipulate a signal via complex weighting of the analysis frames is proved in Section 3) , the cores were accepted into service as research and development tools.

### 2.3 4096-bin Filterbank Acceptance Testing

The firmware applications used in this test procedure have the following designations:

- Filterbank: *rfel\_filterbank\_4096\_250906\_01.ttf*
- Data Capture: *Slot2\_dataCapture\_201106\_02.ttf*

The core was operated at 100MSPS. When performing signal reconstruction tests, the filterbank analysis frame weights were set to unity, affecting a transparent mode of operation.

In accordance with the overflow management requirements established at Section 2.2.4, the weights coefficients were scaled by 0.95 (i.e. the near-unity complex weights were  $125000 + j0$ ) and filter coefficients were scaled by 0.90.

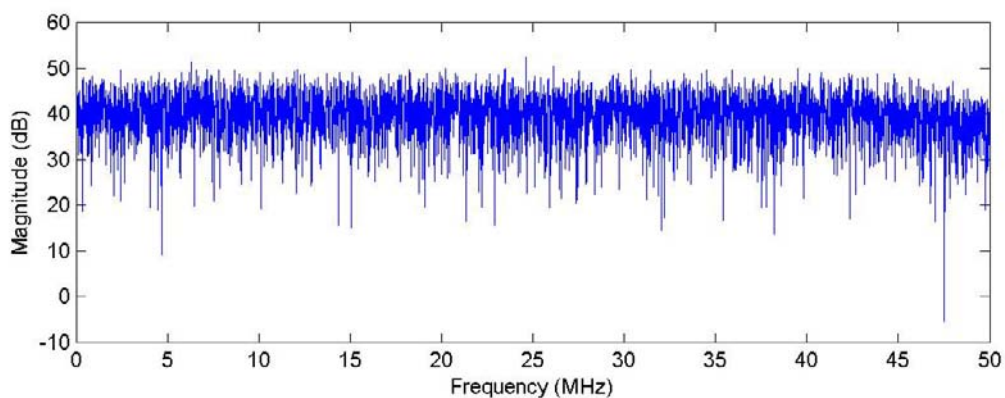
#### 2.3.1 Test Result – Pseudo-random noise #2

A graphical interpretation of the measured signals is given in **Error! Reference source not found.**, and the worst-case EVM is -35.3dB. This is discussed further in the next section.

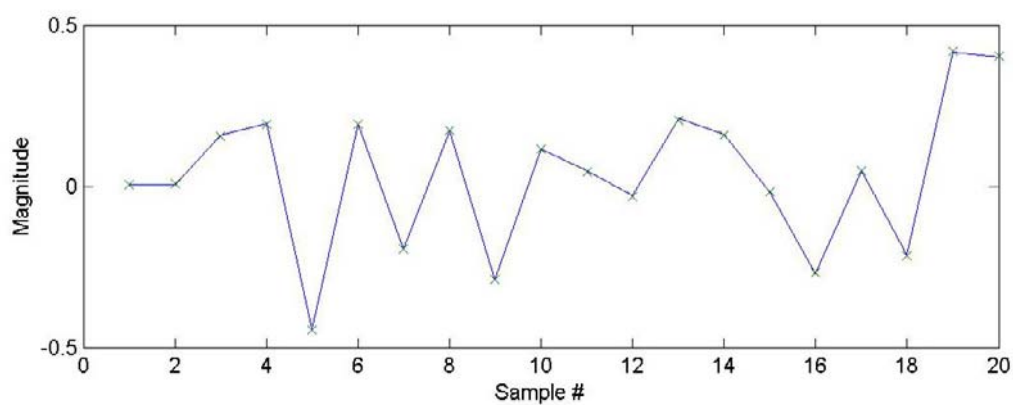
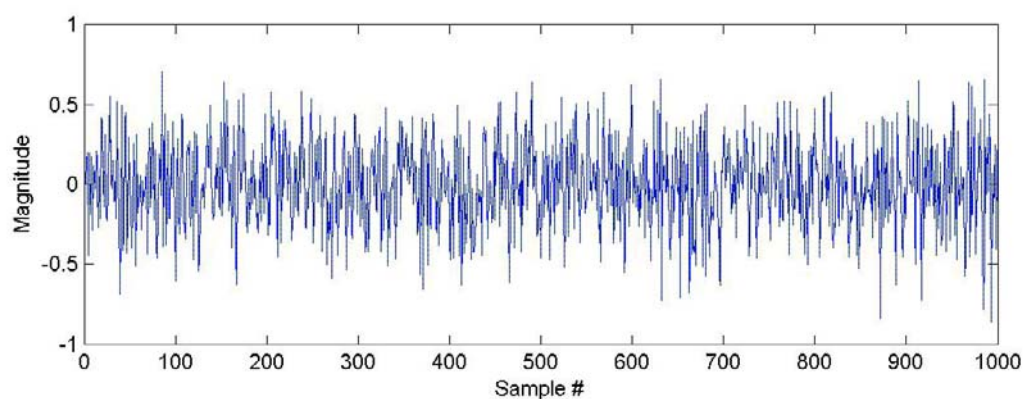
Table 4 - Tests result visualisation – 4096-bin core with pseudo-random noise I/O

**Description:** Test of broadband signal reconstruction, AWGN.

**Input signal power spectrum:**

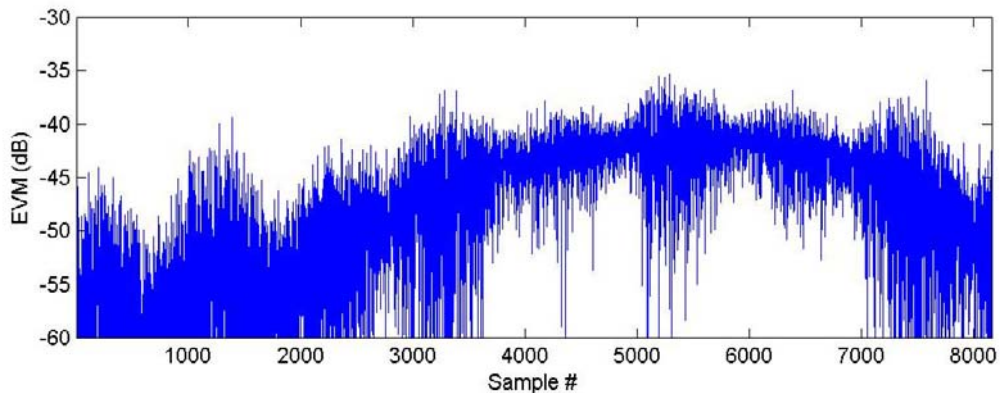


**Typical time domain input vs output:**



**Description:** Test of broadband signal reconstruction, AWGN.

**Typical EVM:**



**Peak EVM:** -35.3dB

### 2.3.2 Test Results – Other cases

For improved readability, other graphical results are segregated to Appendix B: . A summary of the EVM results is given in Table 5.

*Table 5 - Summary of EVM results (4096-bin)*

Stimulus	Comment	Peak EVM
200 kHz tone	Low frequency tone.	-37.1
206 kHz tone	+1/4-bin	-34.4
212 kHz tone	+1/2-bin	-42.9
218 kHz tone	+3/4-bin	-42.9
224 kHz tone	+1-bin	-37.6
1 MHz tone		-43.6
2 MHz tone		-41.6
5 MHz tone		-36.7
10 MHz tone		-37.7
20 MHz tone		-37.0

Stimulus	Comment	Peak EVM
22.857373 MHz tone		-38.0
50 MHz tone		+5.6
Multi-tone, Case #1		-38.8
Multi-tone, Case #2		-37.5
Multi-tone, Case #3		-36.9
Multi-tone, Case #4		-28.2
Stored Square Wave		-44.6
Stored Ramp		-45.1

Reconstruction performance with the 4096-bin core is poorer than that with the 512-bin core in general (the average EVM is -38.7dB, neglecting the 50 MHz tone case). For many applications however, it is expected that this degradation will be functionally insignificant (the -40dB specification equates to a magnitude error of just 1%, and -30dB equates to 3%).

The worst instances of non-compliance occur with the 50 MHz tone and the 4<sup>th</sup> multi-tone stimuli, both of which contain large components near the Nyquist frequency. Additional testing with tones approaching 50 MHz from below showed that there is a deficiency with this core's ability to handle components in the 50 MHz bin.

An input signal comprising a tone on the I (real) channel and broadband noise on the Q (imaginary) channel was input to the DPC. When the tone was at 49.976MHz (one bin width below the maximum), and all weights were scaled to  $W = 0.95 + j0$ , the output was a valid reconstruction of the input, as expected. If the weights for the (positive and negative frequency) bin at 49.976 MHz were nulled ( $W_2 = W_{4096} = 0 + j0$ ), then at the output the tone component was correctly nulled. As the tone frequency increased toward 50.000 MHz, there was an increasing level of 'crosstalk' between the I and Q outputs so that the I output was a proportion of the tone, and the Q output was a superposition of the tone and the noise. The inability to excise the 50 MHz signal and the erroneous crosstalk were not alleviated by either nulling the -50.000 MHz bin ( $W_1 = 0 + j0$ ) or reducing the scaling of weights and filter coefficients.

In the next section, the potential impact of this flaw on practical signals is considered.

### 2.3.3 Confidence testing with the DSSS Demodulator

Refer to Section 2.2.3. The BER results before and after insertion of the 4096-bin filterbank are shown in Figure 7.

These results indicate that there is no appreciable degradation in the demodulator performance when operated in-line with the filterbank, and hence the filterbank's continuous real-time reconstruction performance is validated.

These results also suggest that with practical signals, rather than contrived test tones, the crosstalk flaw identified in section 2.3.2 is unlikely to manifest as appreciable signal degradation.

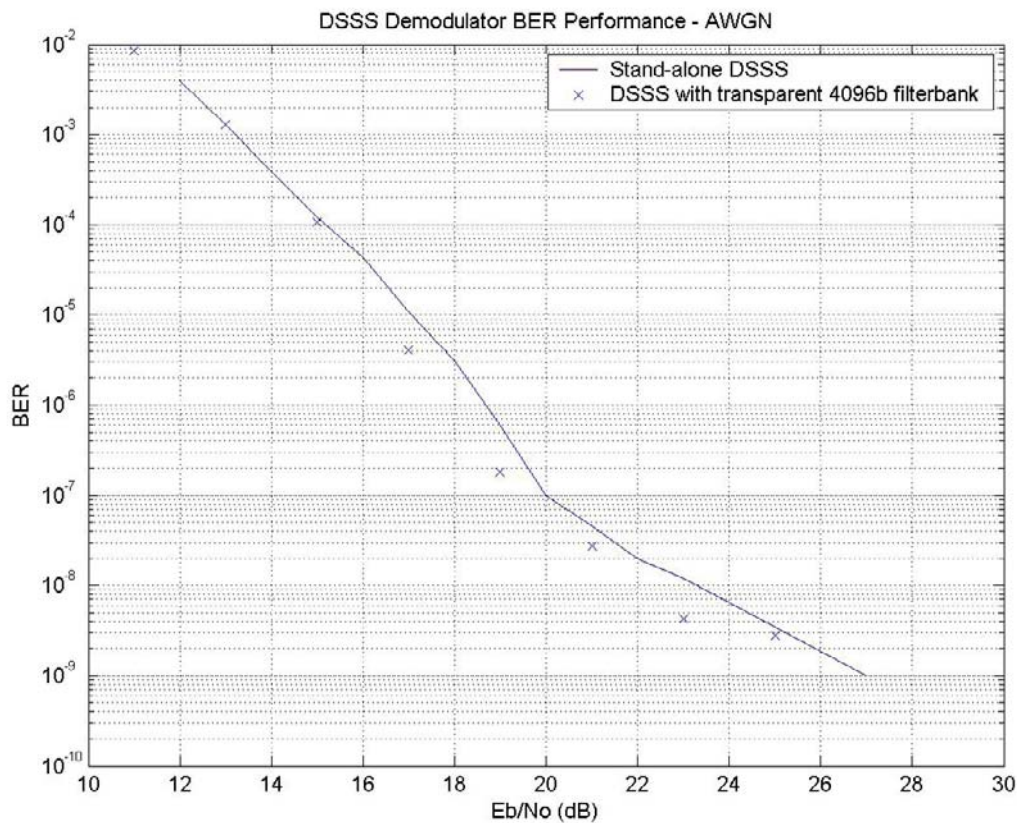


Figure 7 - BER performance of the DSSS demodulator with in-line 4096-bin filterbank

### 2.3.4 Statement of Acceptance

The 4096-bin filterbank IP core does not satisfy the original design requirement for at least -40dB EVM in the reconstructed signal. Nonetheless, with the following concessions the core is accepted:

- The effective EVM for most signals is better than -30dB, and this degradation will still be acceptable in many practical situations.

- RFEL was consulted about this issue within the warranty period of the contract (and hence could be asked to redesign the core). However RFEL was unable to precisely explain the problem, leading to the likelihood that a bug-fix would require substantial effort and redesign, possibly rendering the core unusable within the tight constraints of the target FPGA. It is preferable to retain the capability with a slight degradation.

Note that the as yet untested ability of the cores to correctly manipulate a signal via complex weighting of the analysis frames is proved in Section 3.



### 3. Excision Filter Applications

#### 3.1 Scope and Procedure

The arrangement for excision testing with the DSSS demodulator is shown in Figure 8.

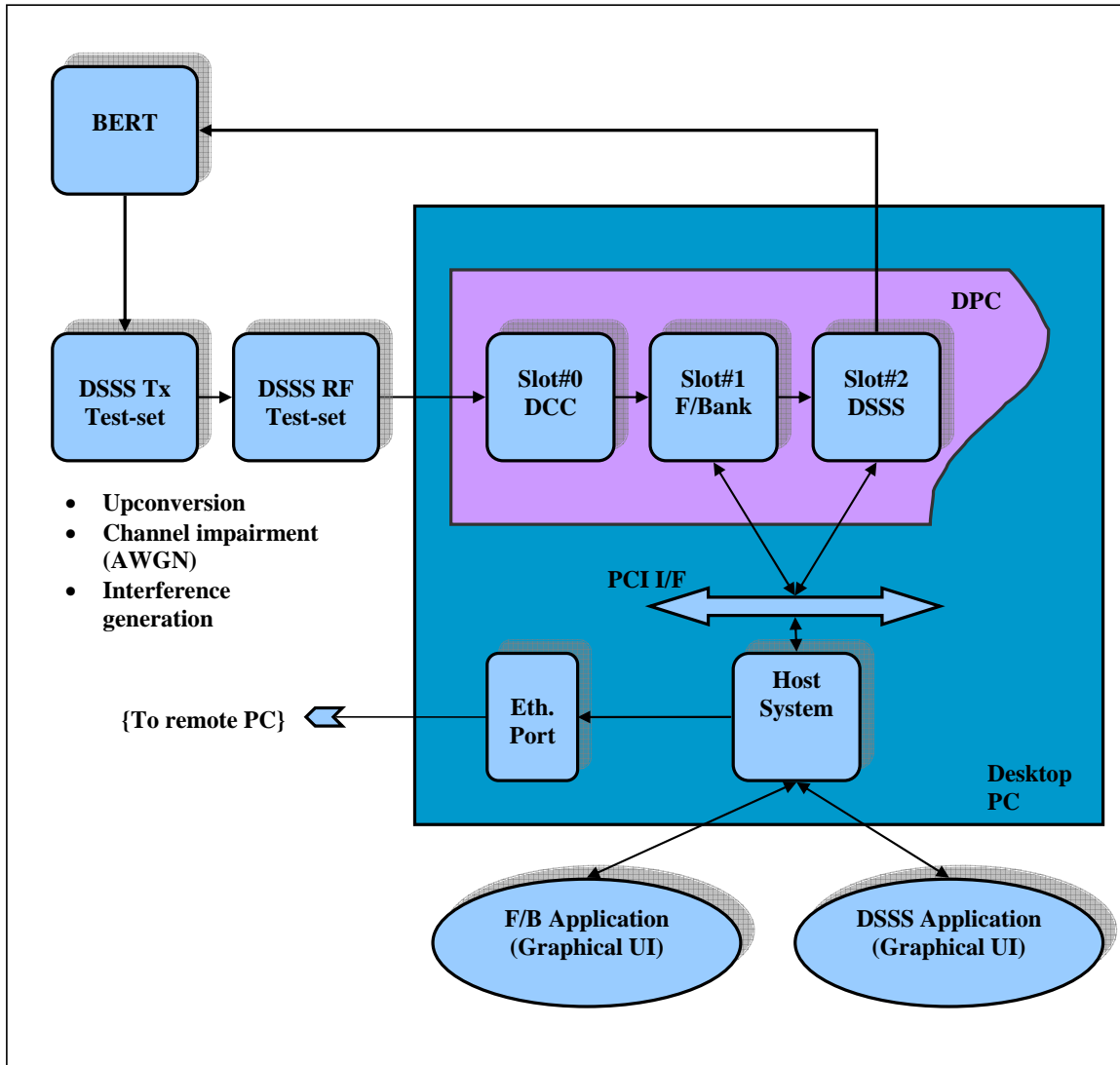


Figure 8 – Test arrangement for excision filtering

The BERT generates a message signal in the form of a pseudo-random binary sequence (PRBS). In these tests:

- The bit rate was 2 Mbps.
- The sequence length was  $2^{31}$ .



This is connected to the DSSS Transmitter (Tx) test set, a collection of test and measurement equipment based around a COTS vector signal generator (VSG). The Tx test set spreads the message, adds a controlled proportion of AWGN to achieve a desired Eb/No operating point (in the spread time domain), and quadrature-modulates this spread, noisy message onto an RF carrier at a set power level. In these tests:

- The spreading rate was 50 MCPS.
- Eb/No was varied from -4dB to +4dB.
- The signal was transmitted at 2040 MHz with power -60dBm at 0dB Eb/No.

The Tx set also generates a second, independent signal to be used as an interferer, and separately manages the baseband signal characteristics, and the RF frequency and power. In these tests:

- The signal was either a tone, transmitted at 2040.5 MHz to 2080 MHz and with relative power level -9dB to +36dB; or
- A multi-carrier 'comb' signal comprising 32 equal-powered tones spaced at regular intervals of 1 MHz across the band centred at 2040 MHz, and with total relative power -9dB to +30dB.

Both test signals are summed together at RF and the resultant signal is transmitted (by wire) to the DSSS RF Test set. The RF test set is a custom wideband downconverter, which provides tuning, AGC and downconversion to baseband.

This baseband signal is suitable for input to the DPC, where it is sampled, optionally filtered in Slot#1 by a filterbank application, then demodulated in Slot#2 by the DSSS Demodulator core. This core outputs a copy of the message signal which is returned to the BERT for comparison with the original message to establish an error-count. In these tests, interferer severity was increased until the BER was of the order of  $10^{-2}$ .

## 3.2 Degradation of the DSSS Demodulator in the presence of Interference

### 3.2.1 Theoretical

The behaviour of a DSSS demodulator in the presence of different types of interferer is discussed in [10]. This prior knowledge assists in planning the scope of the test, assessing the validity of test results, and representing the results.

In general, BER performance is plotted against the independent variable  $10\log_{10}[(P/J)*(W/R)]$  for a range Eb/No operating points and this approach is also taken here. In this expression:

- P = desired signal power;
- J = total jamming signal power;
- W = spread-spectrum (null-to-null) bandwidth; and
- R = bit rate.

In each jamming scenario, the BER is seen to degrade smoothly as  $J$  increases [10, Figures 6-5 (barrage noise), 6-13 (partial band), and 6-23 (single tone)]. For example, Figure 9, for the case of [10, Figure 6-13], is reproduced from Equation 6.31 of the reference (and adjusted for known discrepancies between the modelled demodulator and the DSTO DSSS demodulator), i.e.

$$P_b = Q\left(\sqrt{\frac{2}{R \cdot N_o/P + 2 \cdot (J/P) \cdot (R/W)}}\right)$$

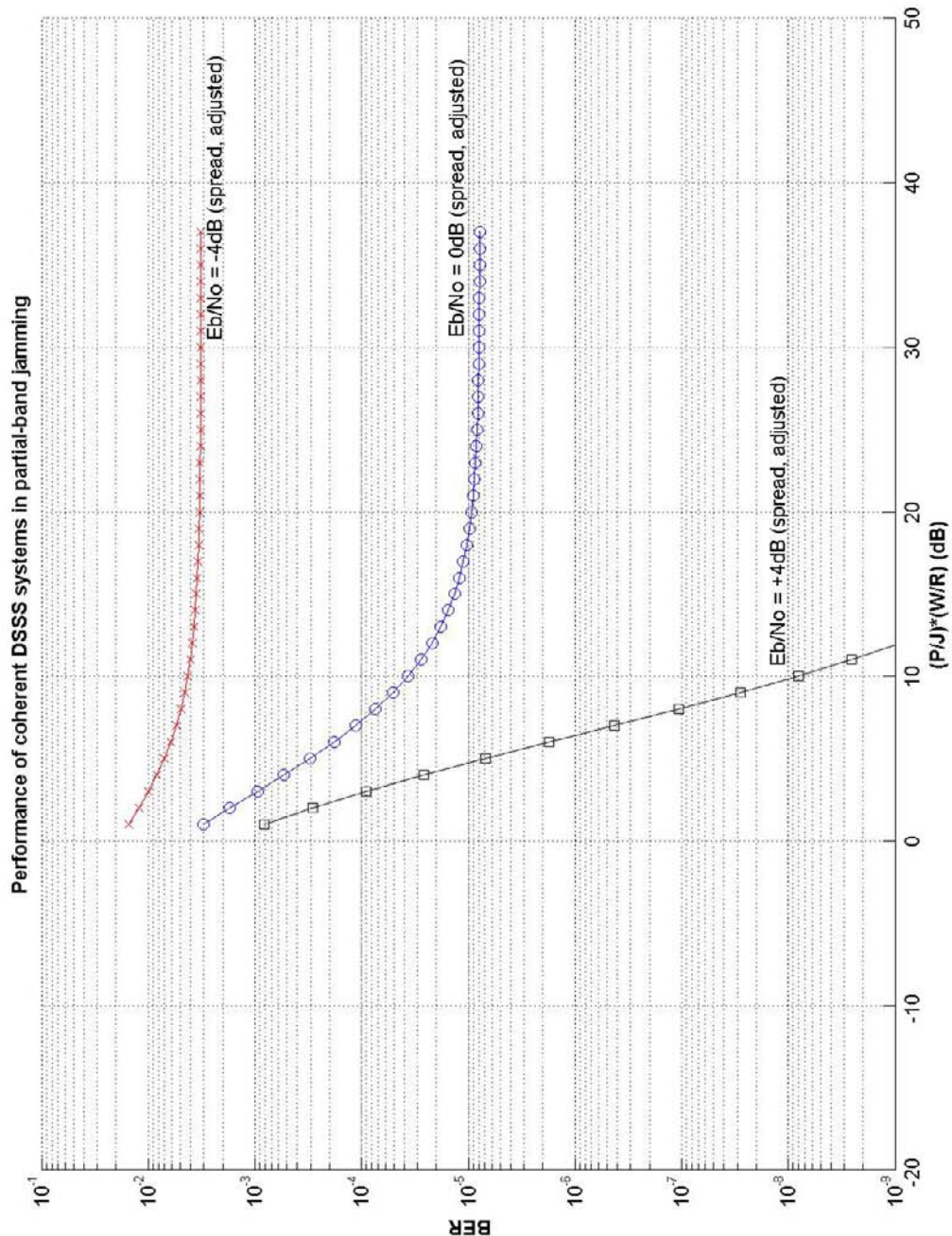


Figure 9 – Theoretical DSSS performance degradation with partial-band jammer

The effect of receiver AGC should also be considered. Although the filterbank may excise an interfering tone, this operation occurs after the analogue signals are sampled and hence the tone is still present at the time of sampling. A practical receiver is likely to have an AGC mechanism which acts to prevent saturation at the sampler, so that if the interference is large enough, the AGC will attenuate the signal prior to sampling, and the desired signal components may fall below the sampler noise floor.

### 3.2.2 Measured

Measured data is logged and analysed in C.2, with the following results.

#### 3.2.2.1 *Single tone jammer versus band position and relative power*

Refer to Figure 10. The ratio (W/R) is 17dB ( $10\log_{10}(100 \text{ MHz}/2 \text{ Mbps})$  ).

At high ratios of P/J, the result tends to the AWGN performance at that operating point. Then as J increases, the error rate increases. Consistent with [10, Figure 6-23], the degradation is worse for a tone near the band centre, tending to negligible near the band edge (at least until the jammer power drives the AGC to attenuate the desired signal).

Some degradation is apparent even at the first measured point, which lies at P/J = +9dB (i.e. when the jammer power is 9dB *less* than the signal power) and additional degradation is rapid. This reveals both the sensitivity of the DSSS waveform to narrowband interference and the potential merit from interference excision.

#### 3.2.2.2 *Multi-tone jammer versus relative power*

Refer to Figure 11. The ratio (W/R) is 17dB ( $10\log_{10}(100 \text{ MHz}/2 \text{ Mbps})$  ).

For comparison, the measurements for the single-tone band-centred interferer of equivalent total power are shown with markers.

It is evident that the total power J of the interference has more impact than the number or distribution of interferers.

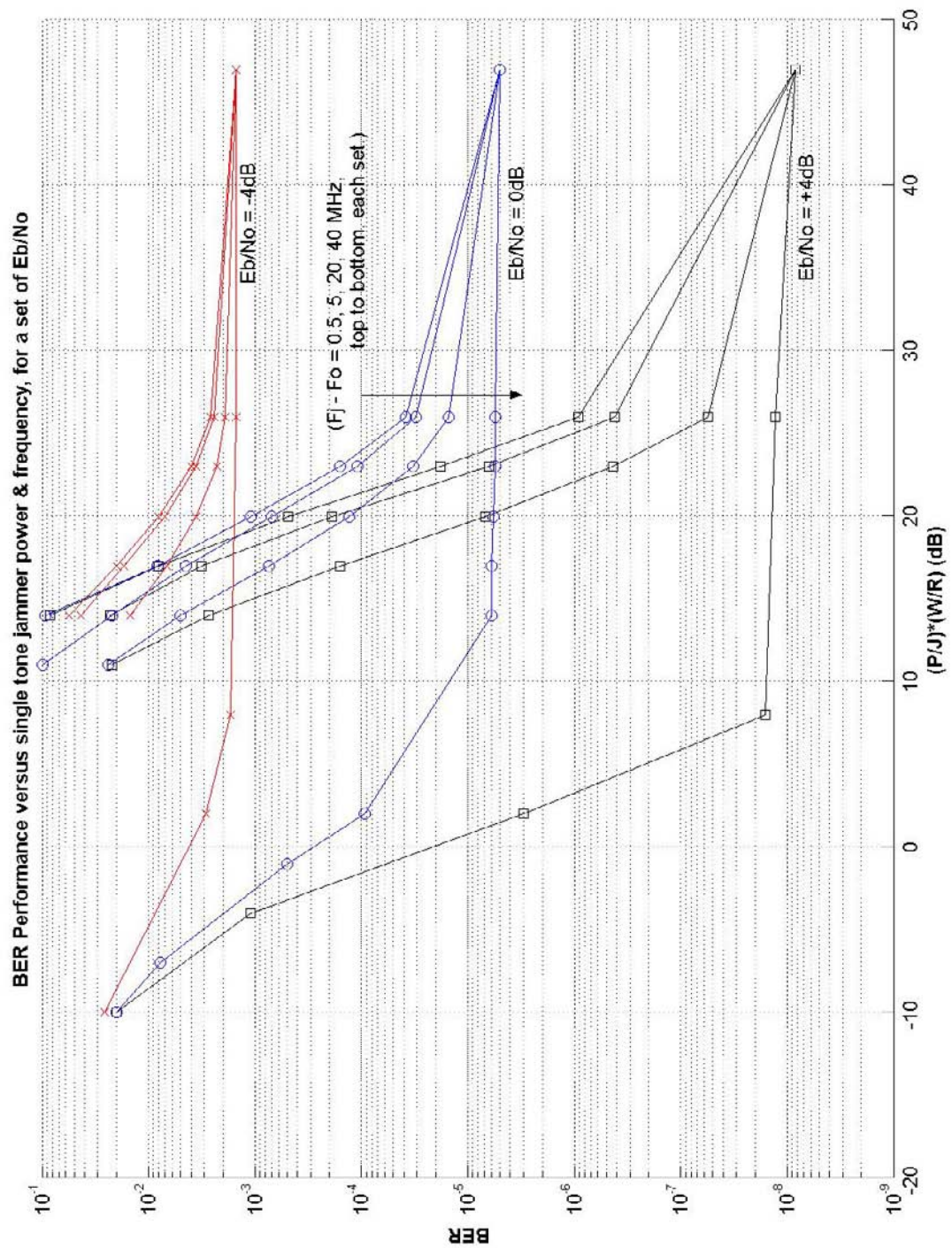


Figure 10 – DSSS performance degradation with single-tone jammer



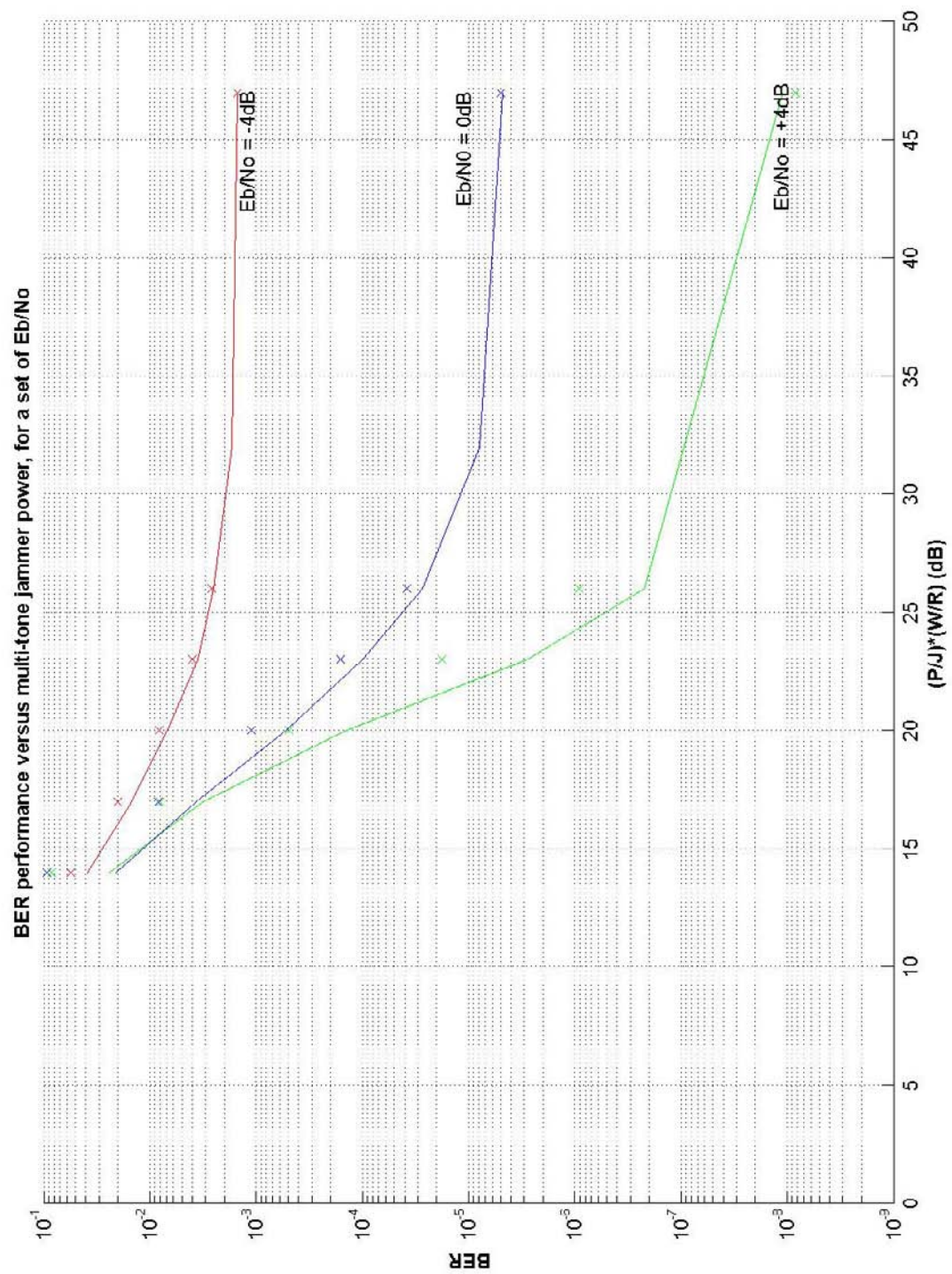


Figure 11 - DSSS performance degradation with multi-tone jammer

### 3.3 Excision Filtering with the 4096-bin Filterbank

Using the Filterbank GUI, the filterbank is readily configured for excision rather than transparent feed-through. For the single- and multi-tone cases identified in the previous section, the filterbank was programmed for excision with manual control of *thresholding* and *bin-width*,  $W_{bin}$ . In this mode, the operator sets a threshold (in dB) on the components of the input power spectrum (estimated by averaging the power in the components of the analysis frame). Any bins with content exceeding that threshold are *excised*, i.e. that bin and its  $W_{bin}$  neighbours on both sides have their complex weights set to  $0+j0$ .

#### 3.3.1 Single-tone Excision

The results for single-tone excision are presented in Figure 12. For comparison, the equivalent results without excision are repeated (dashed lines). There are three features of note:

- The benefit from excision is excellent, with performance restored to almost the pre-interferer level (i.e. the performance as  $[(P/J)*(W/R)]$  tends toward 50dB).
- There is an anomalous case near  $P \approx J$ , where signal levels are increased but are not yet large enough to trigger additional attenuation via AGC. At these operating points, the filterbank is experiencing some numeric overflow, which can be recovered by reducing internal coefficients.
- Performance ultimately degrades with a similar profile to that observed without excision, albeit with a much improved tolerance to  $J$ . At these points, the AGC circuit in the DSSS Rx Test set is attenuating the high interferer power at the expense of the underlying desired signal power.

#### 3.3.2 Multi-tone Excision

The results for multi-tone excision are presented in Figure 13. For comparison, the equivalent results without excision are repeated (dashed lines). Note:

- The benefit from excision is good, but not quite as good as that seen with the single tone case. This is probably because the desired signal is now attenuated in many more locations (see section 3.4).
- The case  $P \approx J$  again introduces overflow in the filterbank, which can be compensated by coefficient scaling.

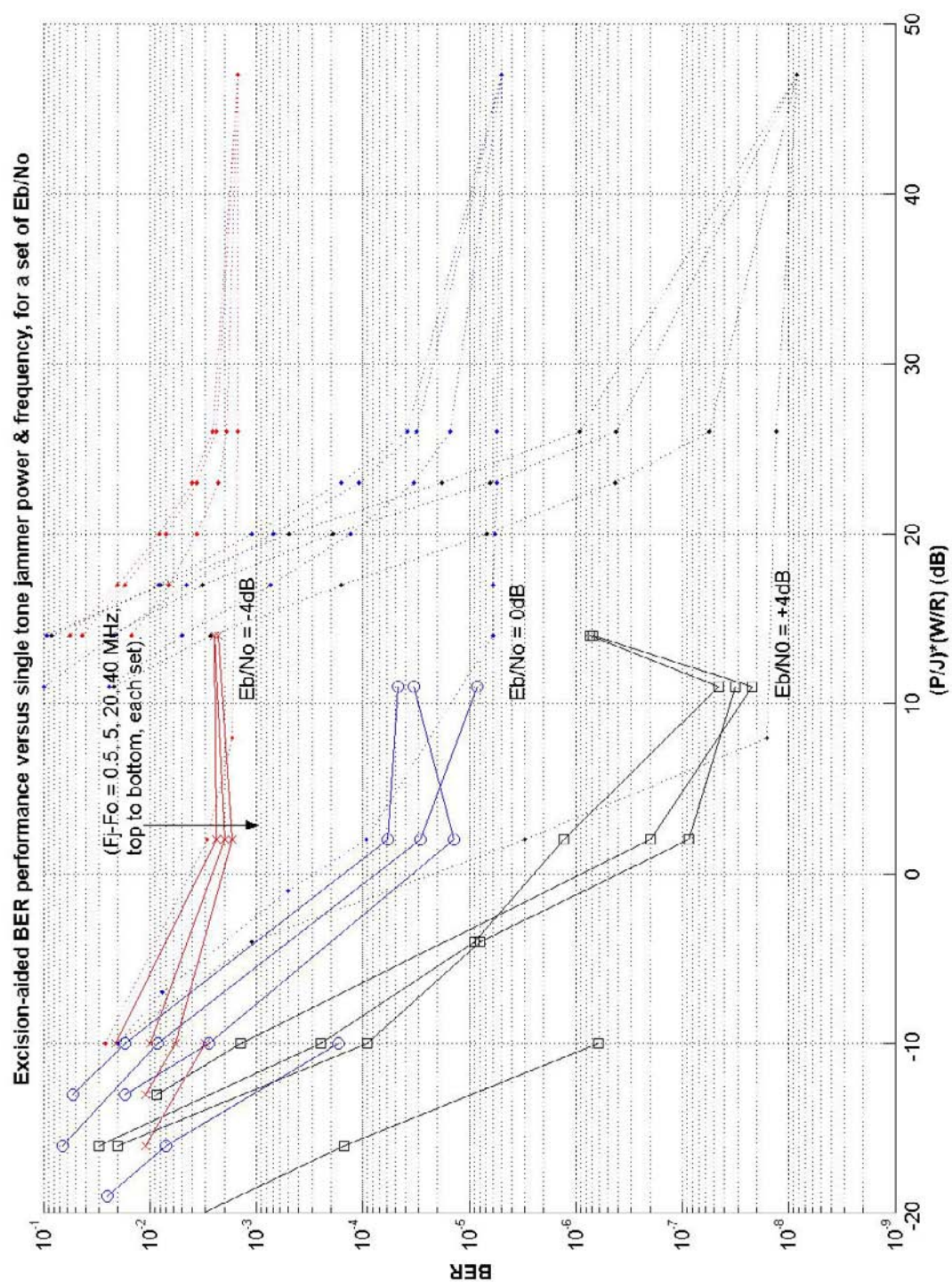


Figure 12 – 4096-bin filterbank excision performance with single-tone jammers



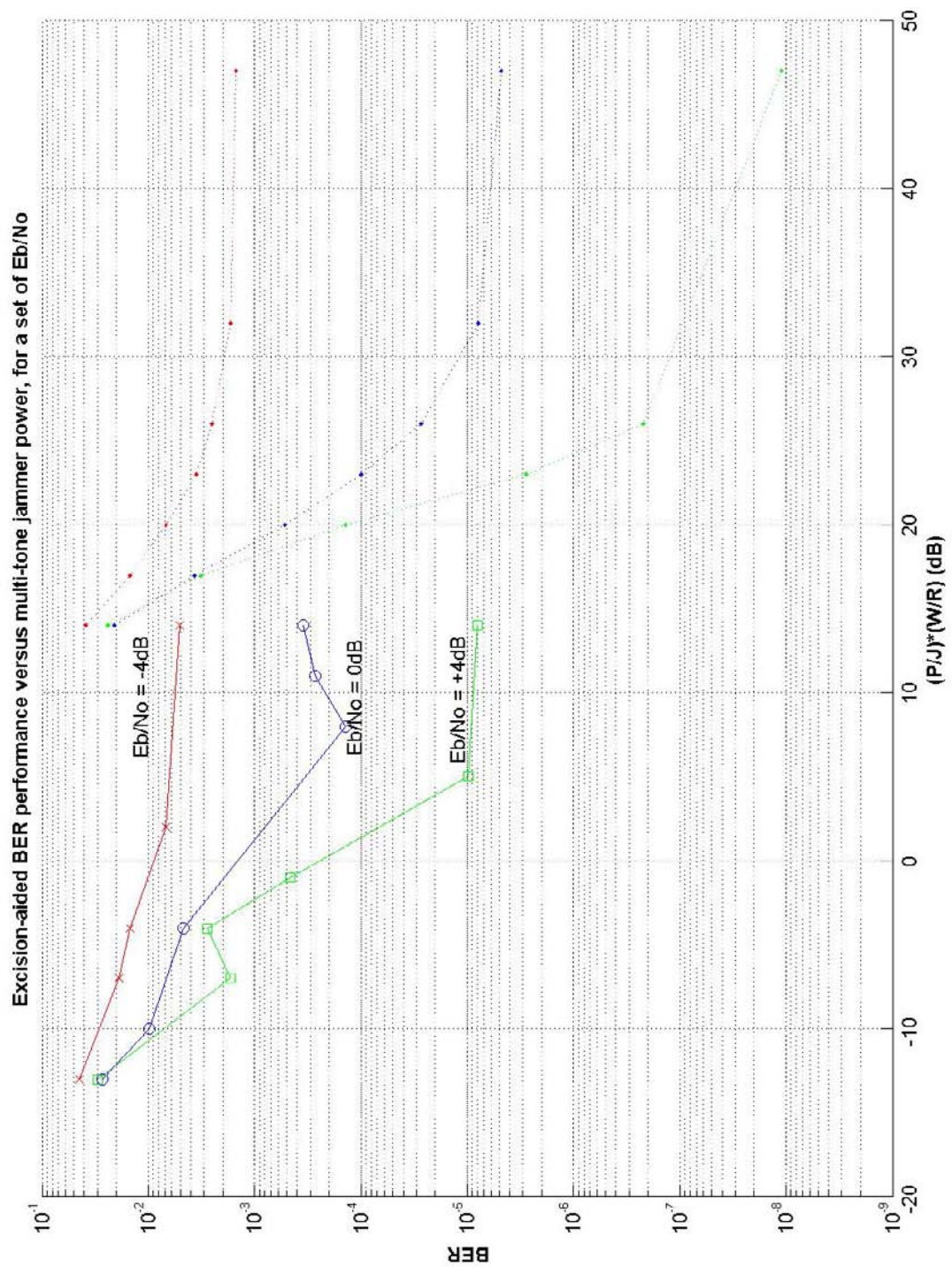


Figure 13 - 4096-bin filterbank excision performance with multi-tone jammers



### 3.4 DSSS Demodulation Degradation in the presence of Bin-nulling

For operating points where the RF Test set AGC is not attenuating the desired signal the performance after excision should be equivalent to that of the interference-free signal with the same number of bins nulled.

Consider two test cases at  $E_b/N_0 = 0\text{dB}$ :

1. Single-tone interferer at 500 kHz offset from band centre:

Table 6 – Comparison between bin-bulling and interference excision – tone jammer

Test	BER
Baseline BER	$6.4\text{e}^{-6}$
BER with jammer at 0dB relative power	$9.9\text{e}^{-3}$
BER with excision enabled (bin width occupied by the interferer is approximately 3-bins)	$9.6\text{e}^{-6}$
BER with no jamming signal but with 9-bins (3 bins at the interferer plus 3 on each side, as is the approach with the previous excision tests) nulled at the same band position	$8.1\text{e}^{-6}$

2. Multi-tone interferer, 32 tones with 1 MHz spacing:

Table 7 – Comparison between bin-bulling and interference excision –multi-tone jammer

Test	BER
Baseline BER	$6.4\text{e}^{-6}$
BER with jammer at 0dB relative power	$3.9\text{e}^{-3}$
BER with excision enabled (bin width occupied by each interferer is approximately 3-bins)	$6.7\text{e}^{-5}$
BER with no jamming signal but with $32 \times 9$ -bin segments nulled at the same band positions	$1.1\text{e}^{-4}$

In both cases, the performance with bin-nulling alone is comparable to that when the interferers are being excised, suggesting that a better filtering algorithm (such as one more closely approximating a Wiener filter) might be able to better recover the baseline performance.

### 3.5 Excision Filtering with the 512-bin Filterbank

Based on the results in section 3.4, the 512-bin application would be expected to perform similarly to the 4096-bin case except with a proportionally worse recovered operating point owing to the increased bin width.

Accordingly, the scope of testing for this case is reduced. The single tone jammer will be considered for the  $E_b/N_o = 0\text{dB}$  case and at 2040.5 MHz only. The multi-tone case will be considered at  $E_b/N_o = 0\text{dB}$  only. Results are presented in Figure 14.

The single-tone case is very similar to those observed with the 4096-bin application. As expected, the wider bins result in slightly more elimination of desired signal and hence slightly worse recovery after excision (at  $0\text{dB } E_b/N_o$ , the 4096-bin core recovers from  $\text{BER} \approx 6.4\text{e}^{-6}$  to a  $\text{BER} \approx 9.6\text{e}^{-6}$ , whereas the 512-bin core yields  $\text{BER} \approx 1.6\text{e}^{-5}$ ).

However the multi-tone case reveals a problem. Here, because the tone spacing is only 1 MHz (5 bin widths), when each tone *and its neighbours* are excised the result is complete annihilation of the desired signal across approximately one third of the main spectral lobe! Demodulation of this signal is difficult.

Note that when the 4096-bin application is excising 32 tones, the number of bins nulled is approximately  $32 \times (3+3+3) = 288$ , which is 7% of the total. The proportional loss for the 512-bin case is 36 bins, which would be realised with just 4 interferers spaced across the band. A test case with four tones at 8 MHz displacements and  $+6\text{dB J/P}$  (at x-coordinate 11dB in Figure 14) resulted in  $\text{BER} \approx 6.9\text{e}^{-2}$  before excision and  $\text{BER} \approx 6.5\text{e}^{-5}$  after, which is more consistent with Figure 13.

### 3.6 Conclusions

Consistent with established literature and intuition, the DSTO DSSS demodulator performance will degrade in the presence of interferers as the ratio of the desired signal power to total jamming power decreases.

The filterbank cores provided by RFEL can be utilised for a coarse bin-nulling excision approach to mitigation against narrowband interferers. Over a wide operating range, this approach recovers an operating point consistent with the performance prior to interference, less the loss of desired signal energy from the excised bins. Consequently, the finer the bin resolution the better the performance of this approach, and the 4096-bin variant is recommended for all applications where its lower sampling rate is sufficient. Care should be taken to ensure that the filterbanks are operated with 'backed-off' scaling to avoid internal numeric overflow.

Practically, all sampling receivers will likely be limited by an AGC circuit, included to prevent saturation of the sampling ADCs. Thus the performance after excision will ultimately be limited by the AGC driving the desired signal below the ADC noise floor.

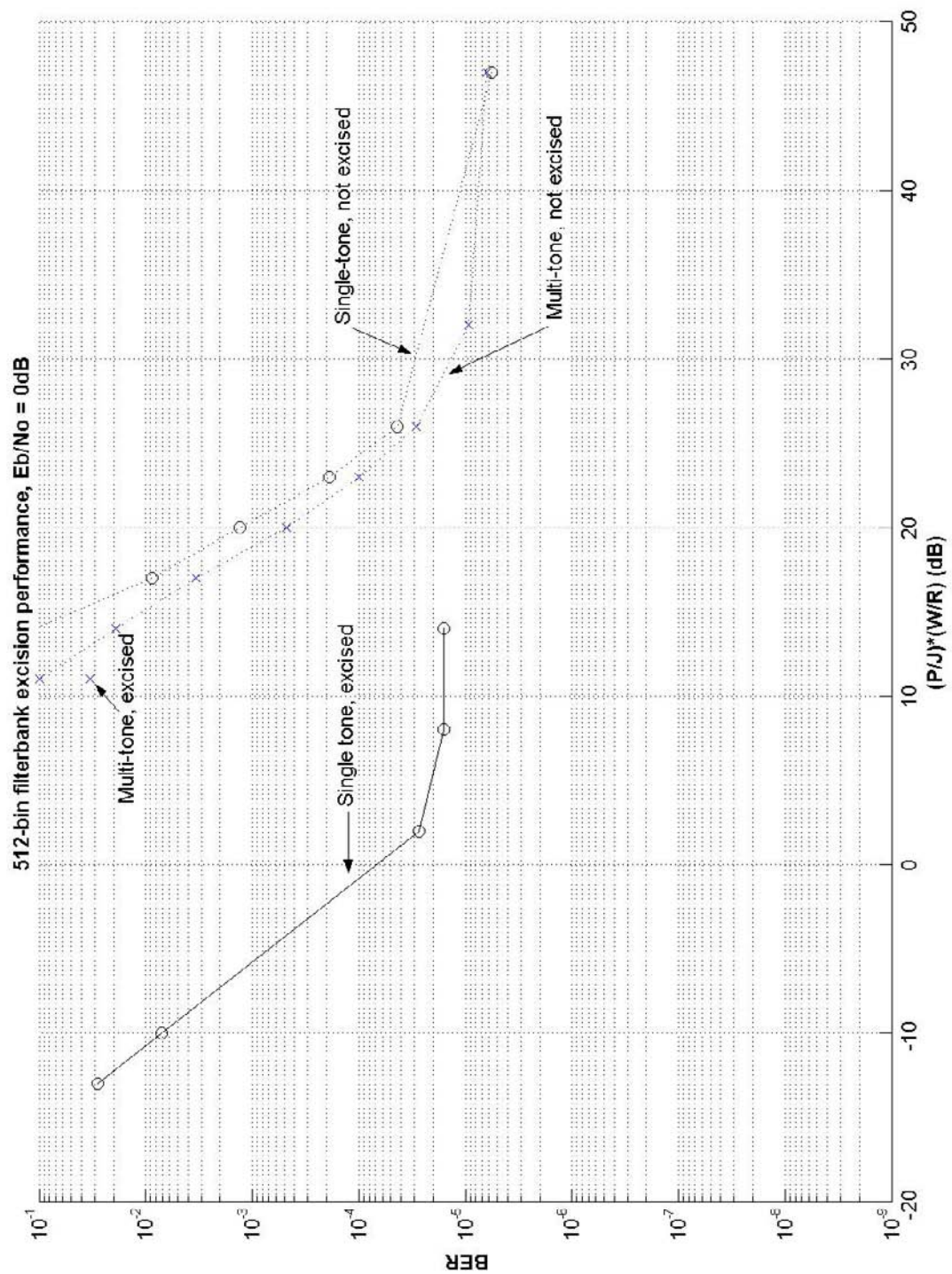


Figure 14 – 512-bin filterbank excision performance with single- and multi-tone jammers

## Acknowledgements

The authors would like to thank Mr Luke Quinane, for his development of the filterbank graphical user interface tools, and Dr Gareth Parker, for his development of the design specification and theoretical discussions concerning frequency domain signal processing.

## 4. References

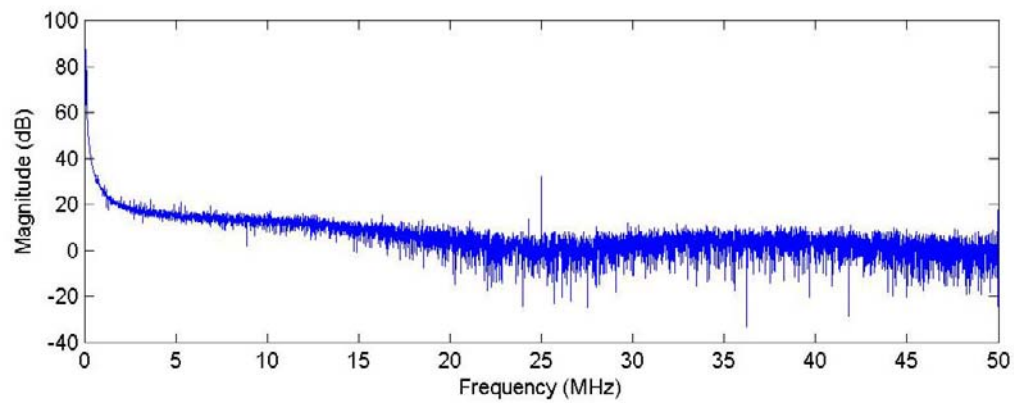
1. RF Engines Limited, "Synthesis and Analysis Filterbank Core for DSTO", RF Engines Limited, March 2005.
2. Harman, K. L., Arnold, J. P., "Wideband Burst DSSS Receiver Program: Phase I Overview", DSTO Technical Report DSTO-TR-(TBA), DSTO, 2007.
3. DSTO, "Contract No. 4500471939 - Provision of a Synthesis and Analysis Filter Bank Core for FPGA", DSTO, 2005.
4. Crochiere, Rabiner, "Multirate Digital Signal Processing", Prentice-Hall, 1983.
5. Ferrara, E. R., "Frequency Domain Adaptive Filtering", Prentice-Hall, 1985.
6. RF Engines Limited, "DSTO Design Study Report", RF Engines Limited, July 2005.
7. RF Engines Limited, "DSTO Interface Specification", RF Engines Limited, December 2005.
8. Harman, K. L., "DSSS Receiver User's Manual", DSTO General Document DSTO-GD-0525 014-083, DSTO, 2007.
9. Harman, K. L., Caldow, A. J., "DSSS Receiver Project Test Report : Initial Characterisation Phase", DSTO Project Document, 2003.
10. Peterson, R. L., Ziemer, R. E., Borth, D. E., "Introduction to Spread Spectrum Communications", Prentice-Hall, 1995.
11. Mathworks, "MATrix LABoratory Software", <http://www.mathworks.com>.
12. Parker, G., "Frequency Domain Restoration of Communications Signals", Institute for Telecommunications Research, University of South Australia, 2001.

## Appendix A: 512-bin Filterbank Acceptance Test Results

### A.1. 100 kHz Tone

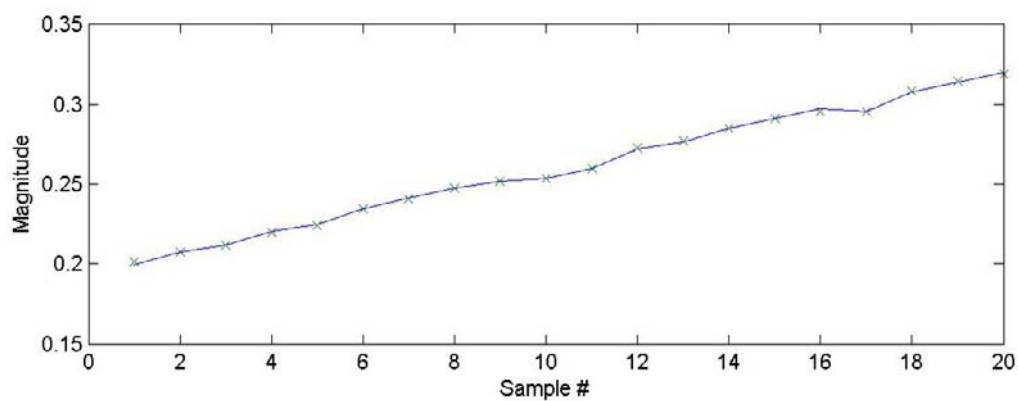
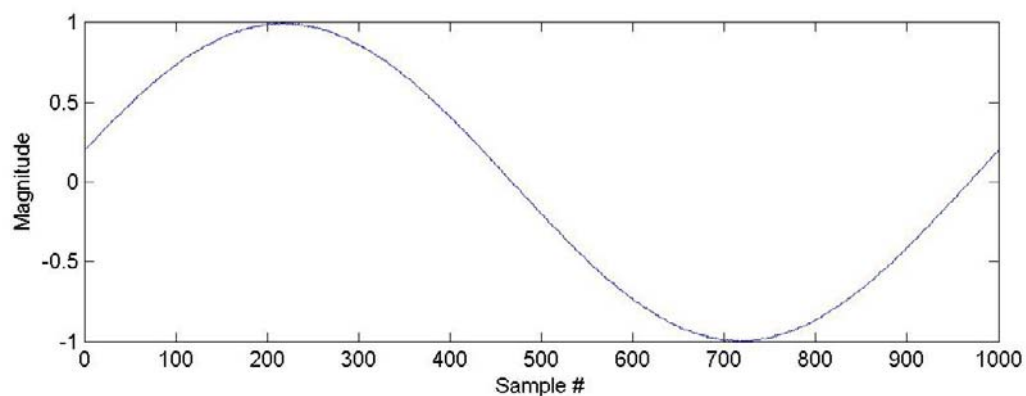
**Description:** Test of low-frequency tone reconstruction, 100 kHz.

**Input signal power spectrum:**

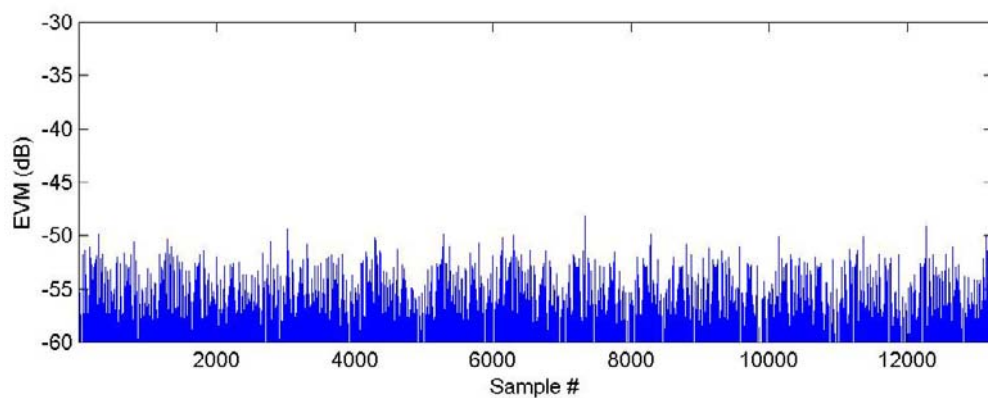


**Description:** Test of low-frequency tone reconstruction, 100 kHz.

**Typical time domain input vs output:**



**Typical EVM:**

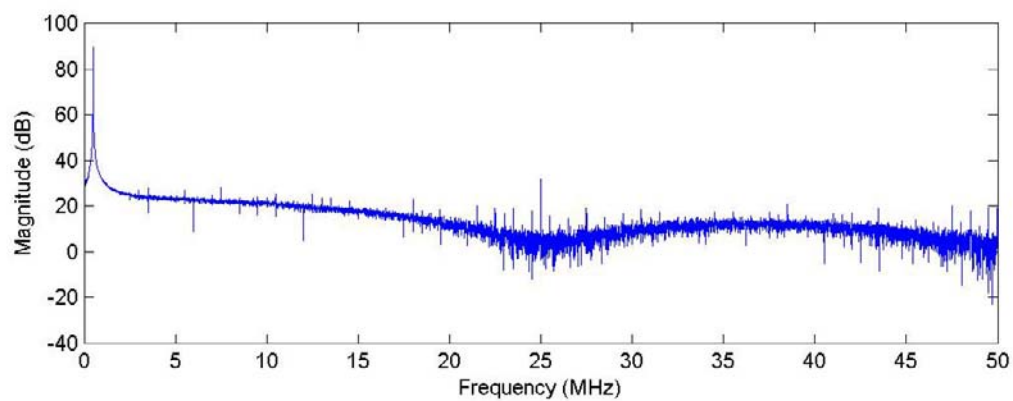


**Peak EVM:** -48.1dB

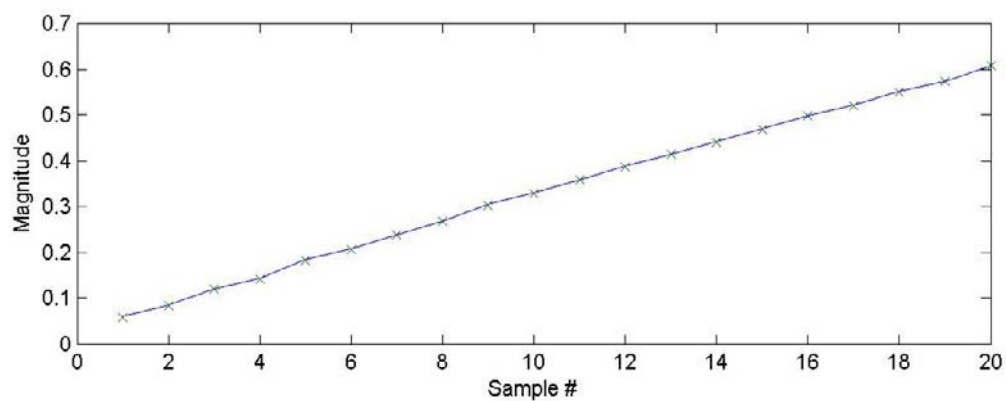
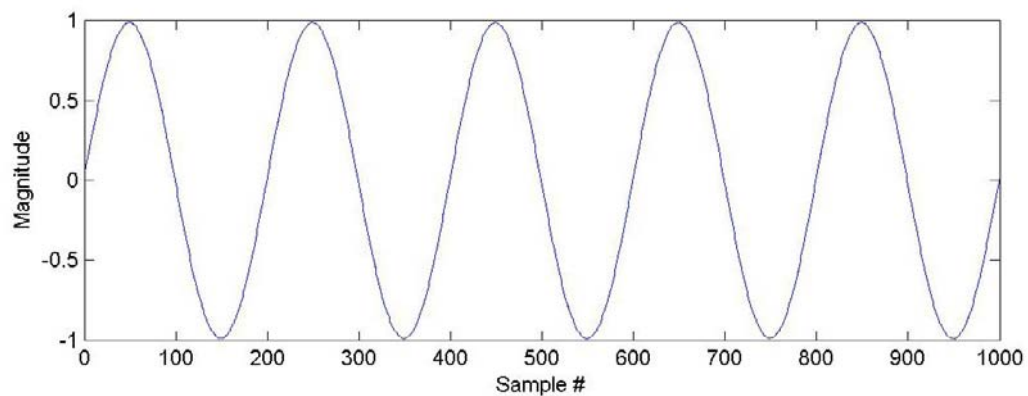
## A.2. 500 kHz Tone

**Description:** Test of low-frequency tone reconstruction, 500 kHz.

**Input signal power spectrum:**

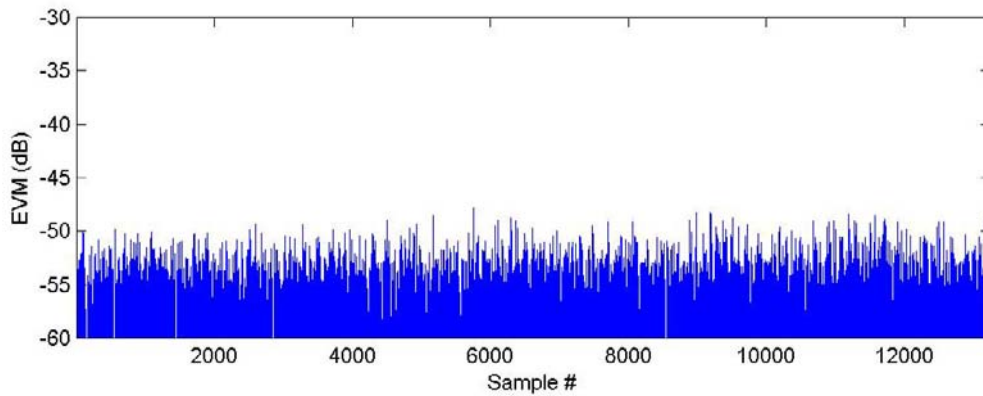


**Typical time domain input vs output:**



**Description:** Test of low-frequency tone reconstruction, 500 kHz.

**Typical EVM:**

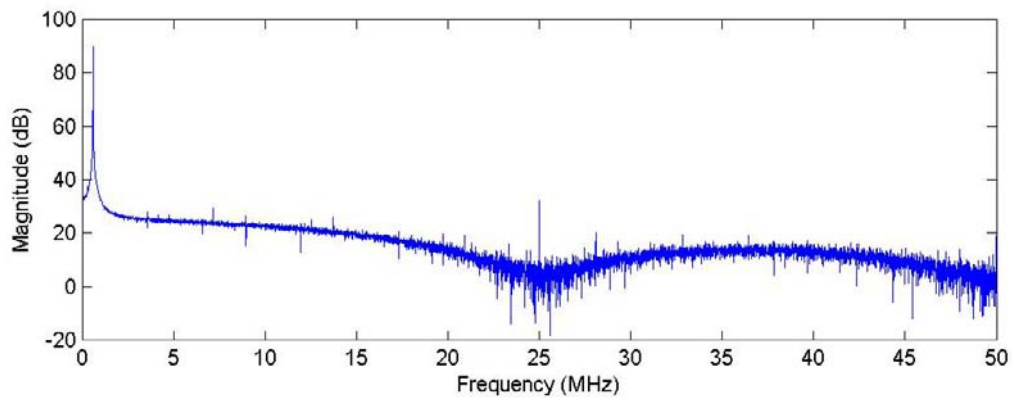


**Peak EVM:** -47.7dB

### A.3. 597.6 kHz Tone

**Description:** Test of low-frequency tone reconstruction, half bin-width offset (positive).

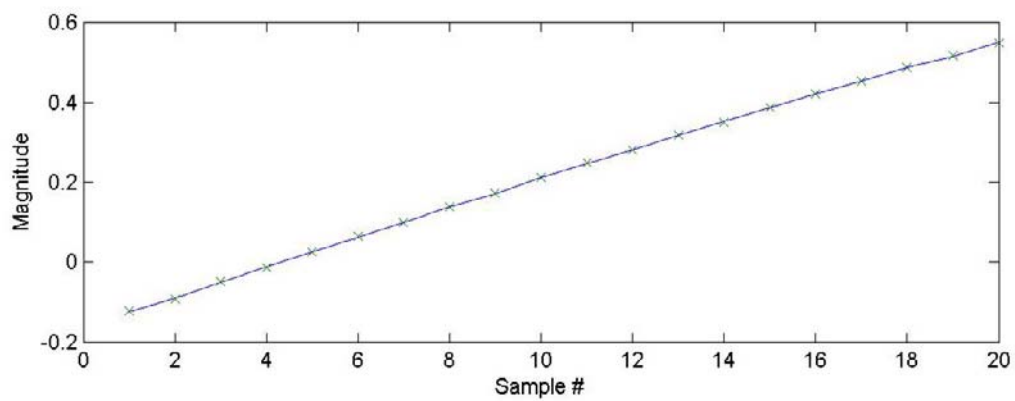
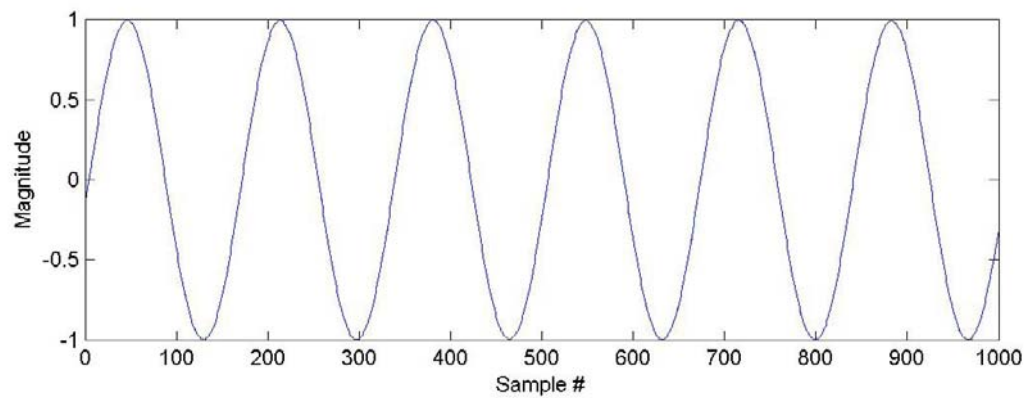
**Input signal power spectrum:**



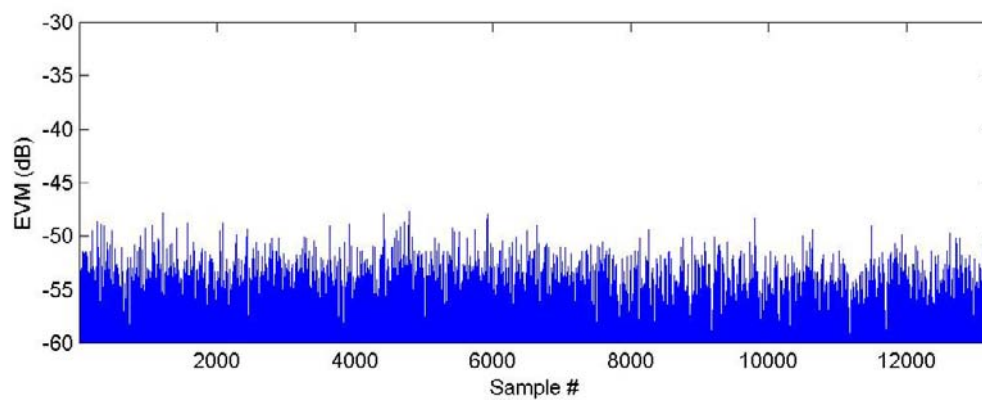


**Description:** Test of low-frequency tone reconstruction, half bin-width offset (positive).

**Typical time domain input vs output:**



**Typical EVM:**

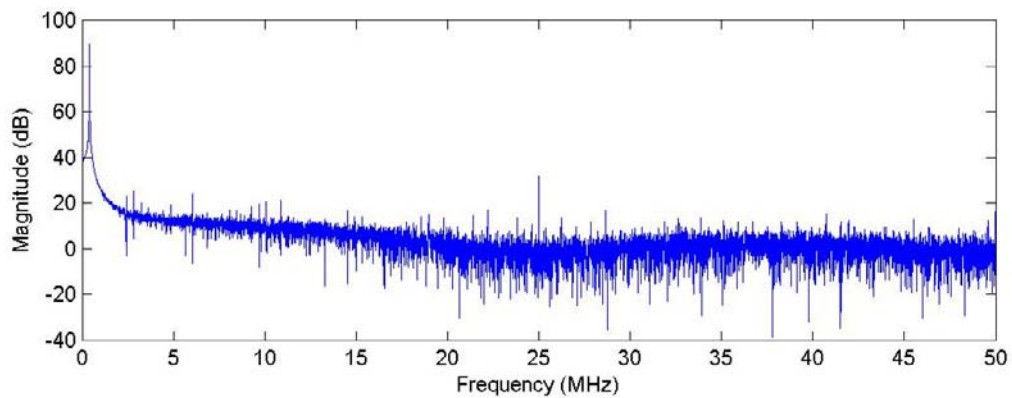


**Peak EVM:** -46.2dB

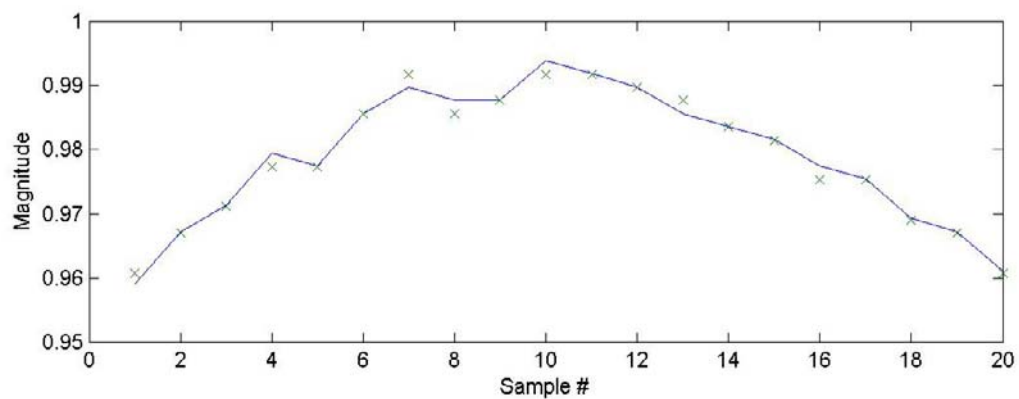
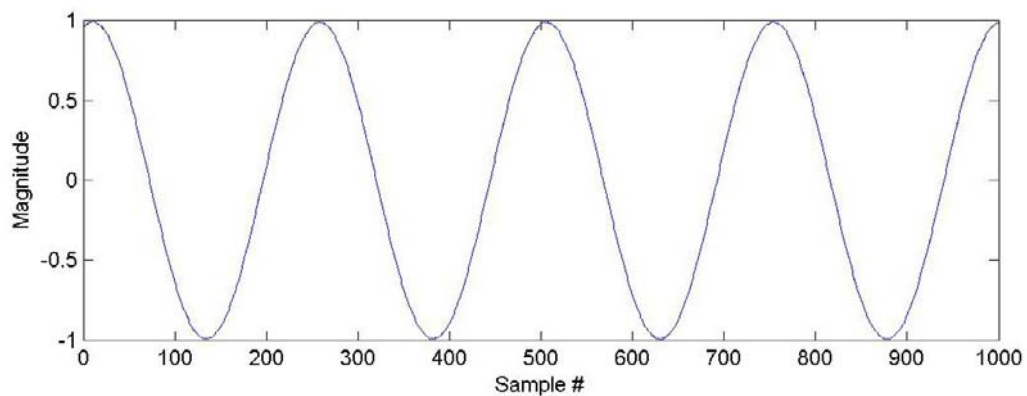
#### A.4. 402.4 kHz Tone

**Description:** Test of low-frequency tone reconstruction, half bin-width offset (negative).

**Input signal power spectrum:**

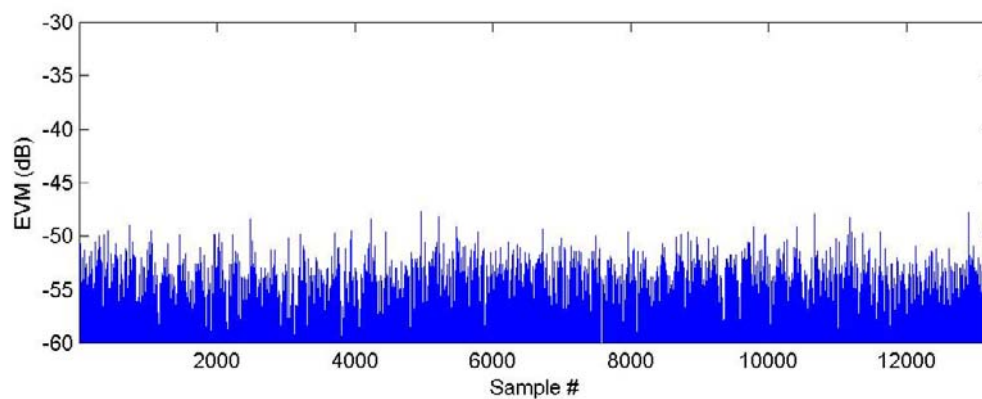


**Typical time domain input vs output:**



**Description:** Test of low-frequency tone reconstruction, half bin-width offset (negative).

**Typical EVM:**

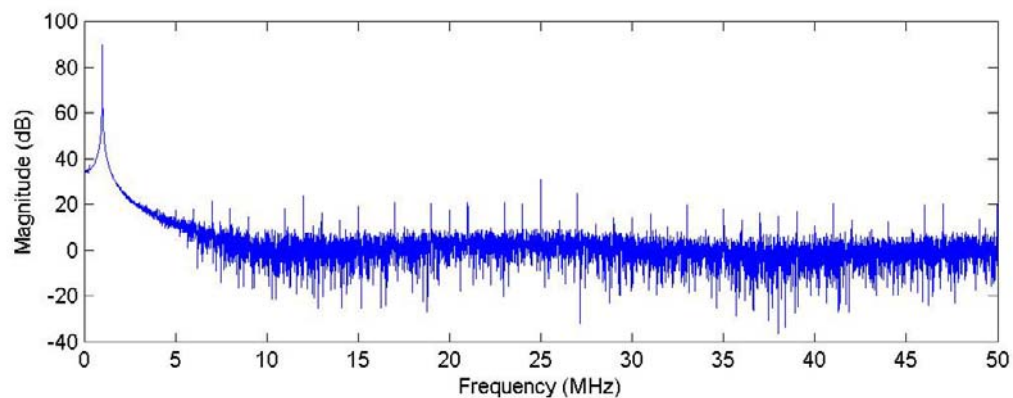


**Peak EVM:** -47.3dB

### A.5. 1.000 MHz Tone

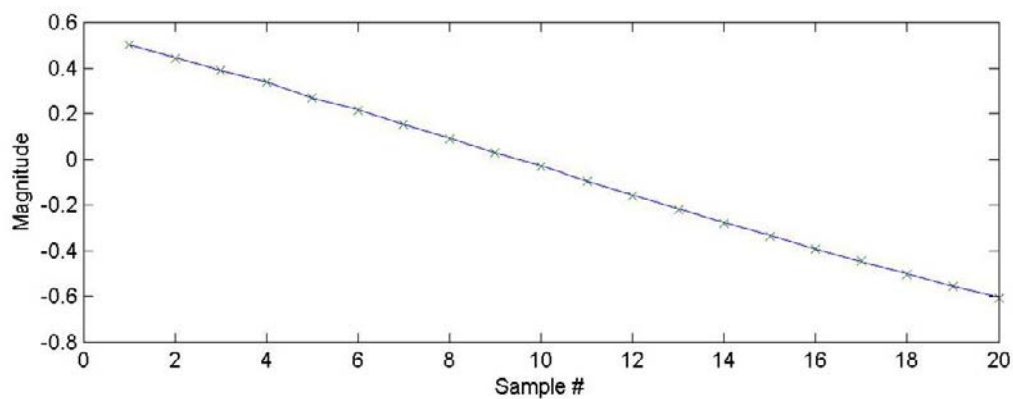
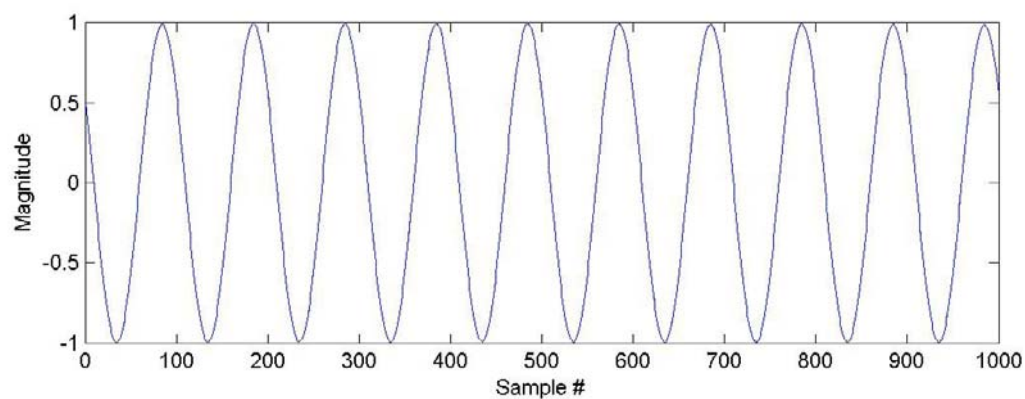
**Description:** Test of in-band tone reconstruction, 1 MHz.

**Input signal power spectrum:**

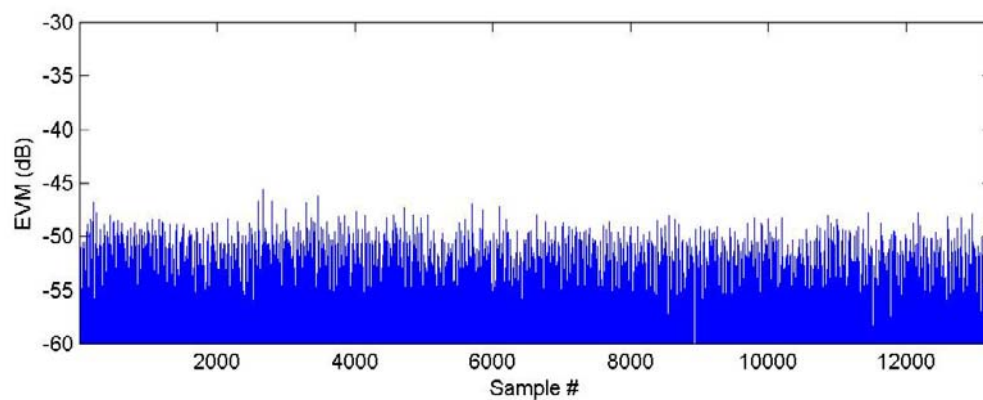


**Description:** Test of in-band tone reconstruction, 1 MHz.

**Typical time domain input vs output:**



**Typical EVM:**

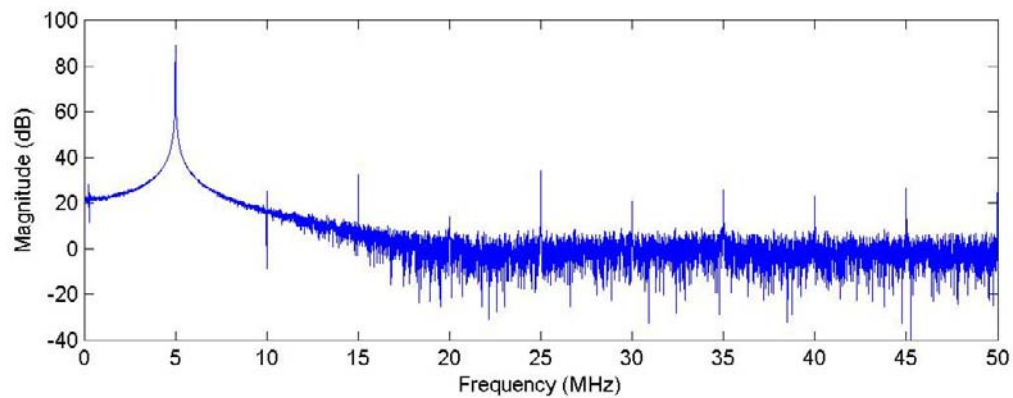


**Peak EVM:** -45.5dB

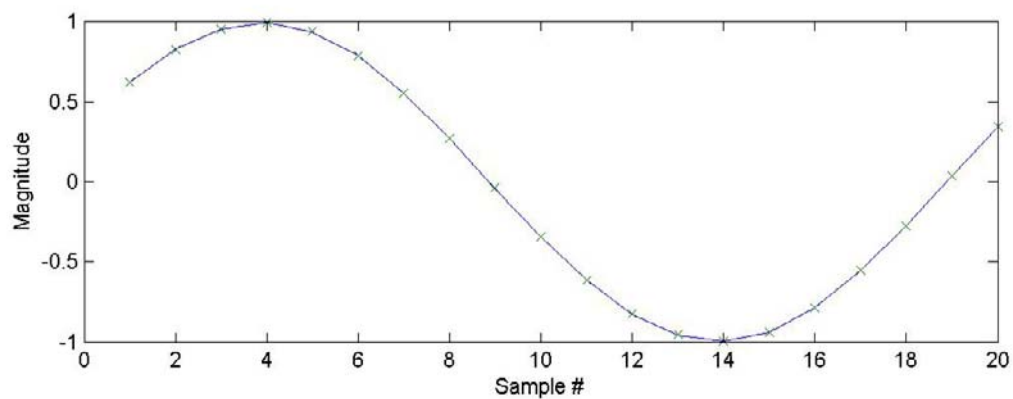
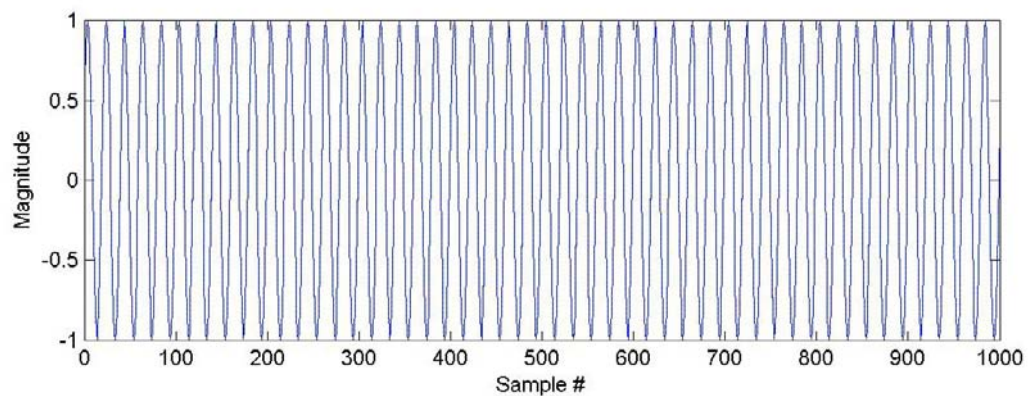
## A.6. 5.000 MHz Tone

**Description:** Test of in-band tone reconstruction, 5 MHz.

**Input signal power spectrum:**

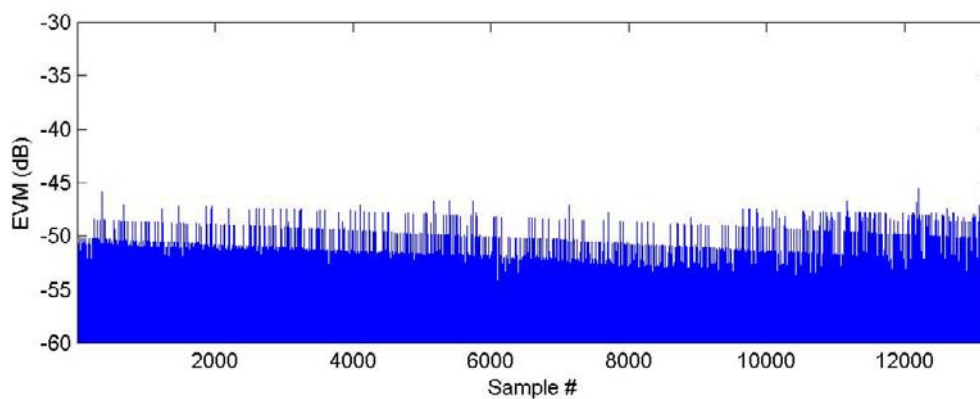


**Typical time domain input vs output:**



**Description:** Test of in-band tone reconstruction, 5 MHz.

**Typical EVM:**

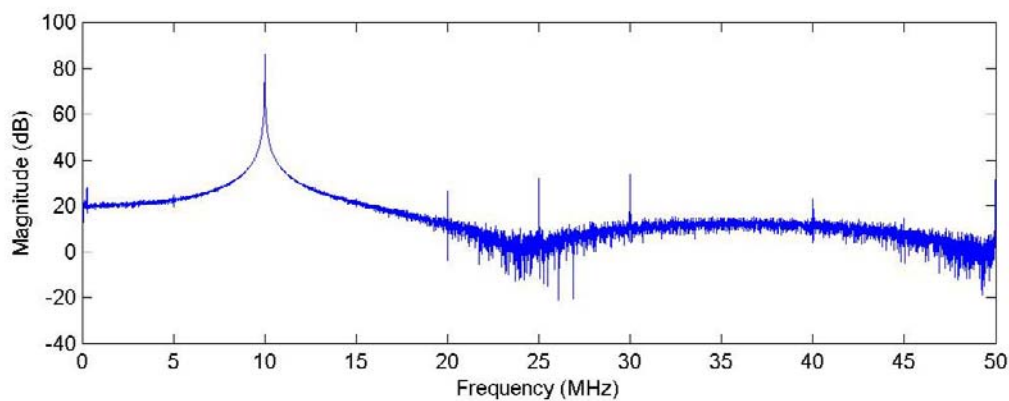


**Peak EVM:** -45.5dB

### A.7. 10.000 MHz Tone

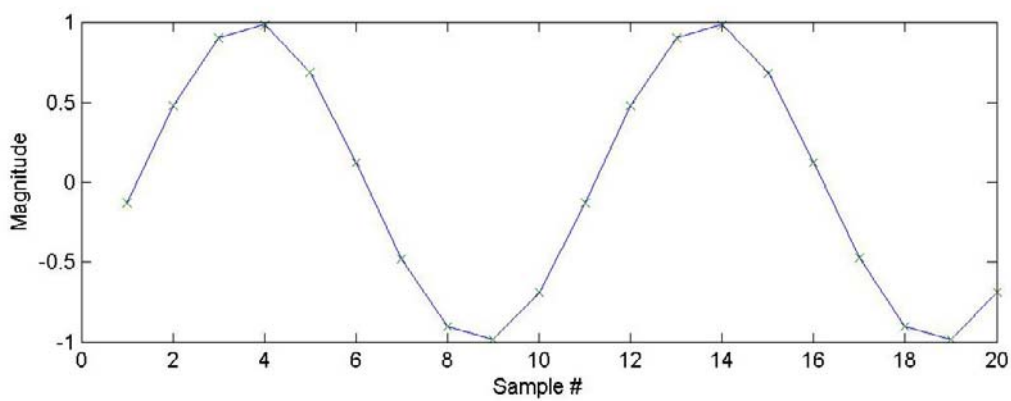
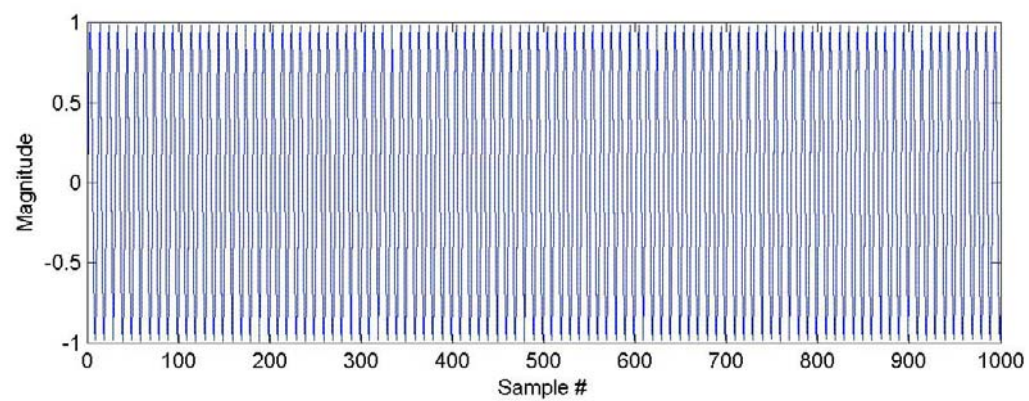
**Description:** Test of in-band tone reconstruction, 10 MHz.

**Input signal power spectrum:**

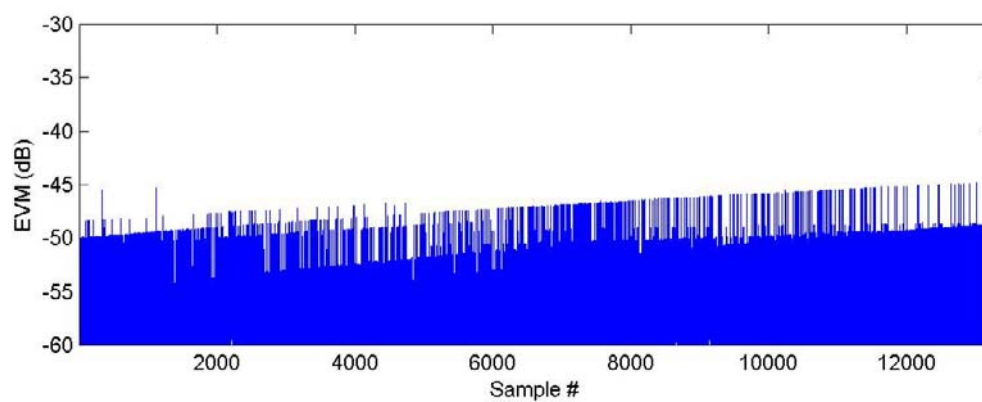


**Description:** Test of in-band tone reconstruction, 10 MHz.

**Typical time domain input vs output:**



**Typical EVM:**



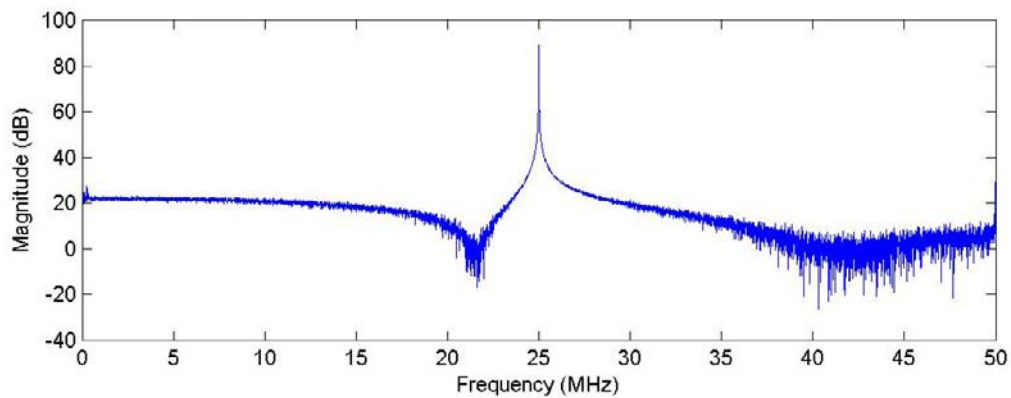
**Peak EVM:** -44.2dB



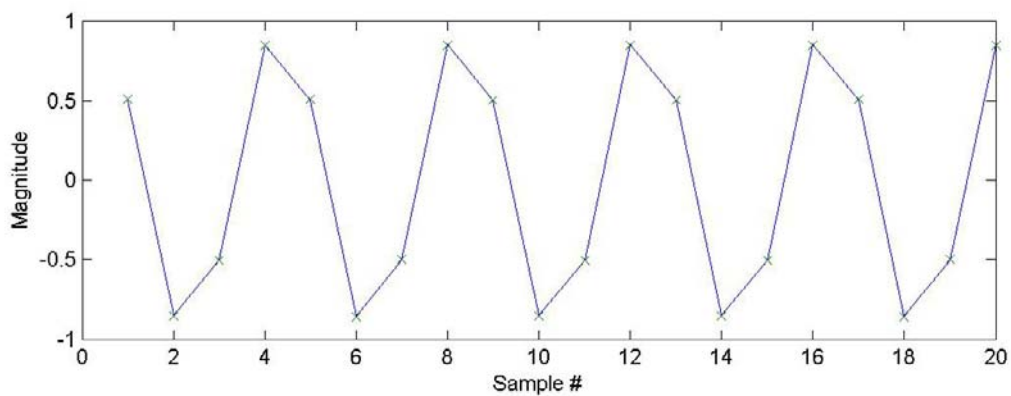
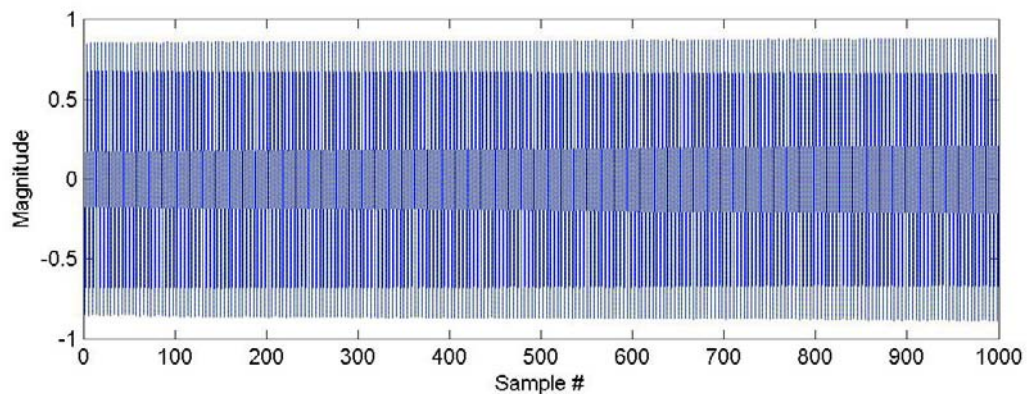
## A.8. 25.000 MHz Tone

**Description:** Test of in-band tone reconstruction, 25 MHz.

**Input signal power spectrum:**



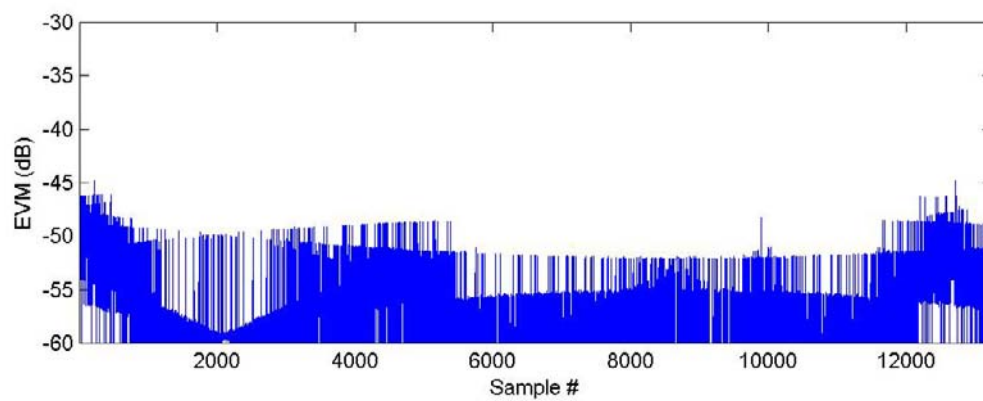
**Typical time domain input vs output:**





**Description:** Test of in-band tone reconstruction, 25 MHz.

**Typical EVM:**

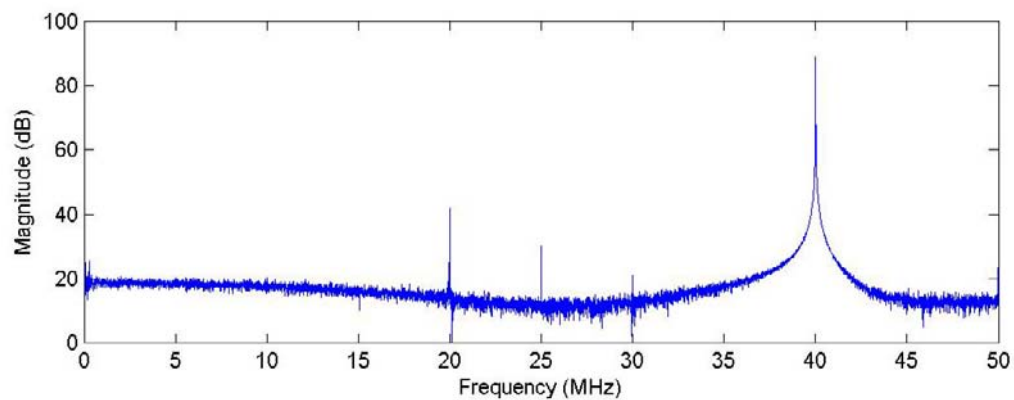


**Peak EVM:** -44.7dB

## A.9. 40.000 MHz Tone

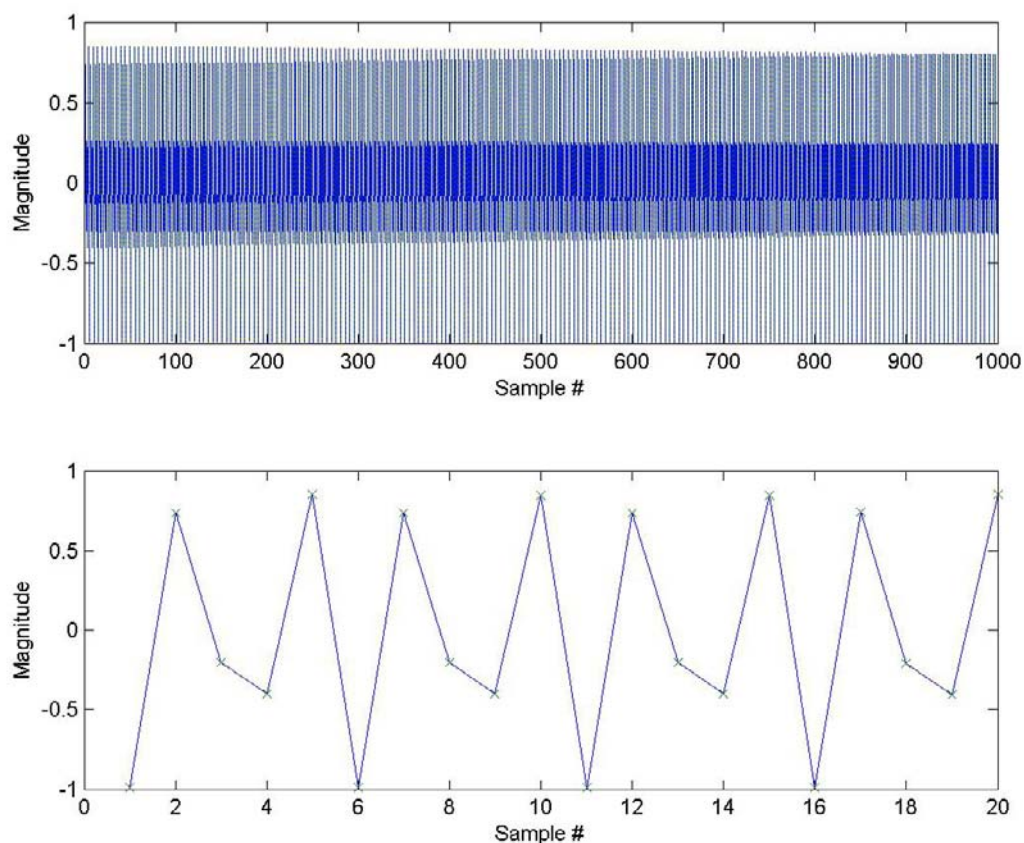
**Description:** Test of high frequency tone reconstruction, 40 MHz.

**Input signal power spectrum:**

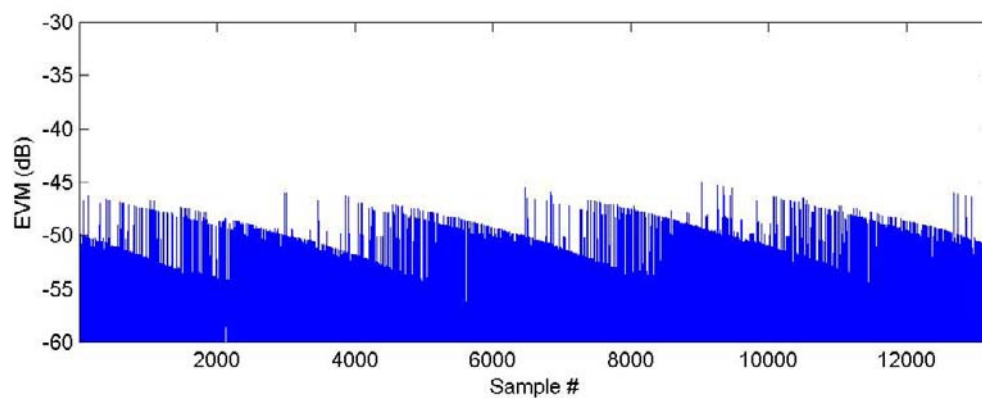


**Description:** Test of high frequency tone reconstruction, 40 MHz.

**Typical time domain input vs output:**



**Typical EVM:**

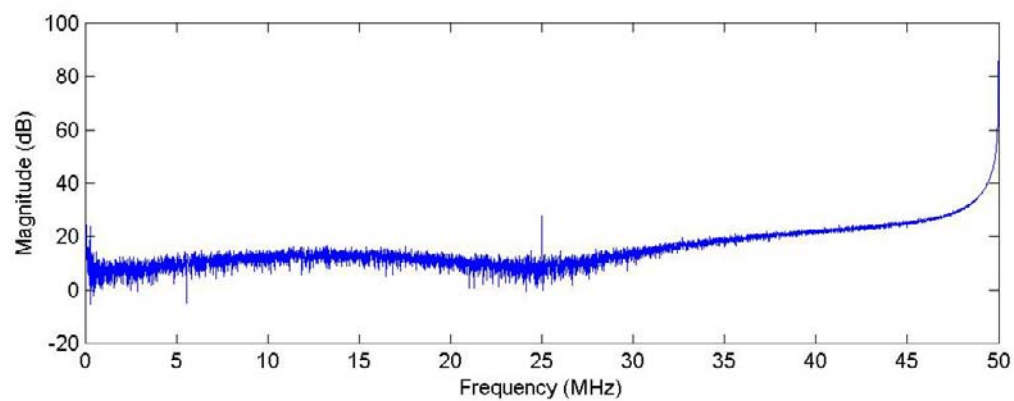


**Peak EVM:** -44.9dB

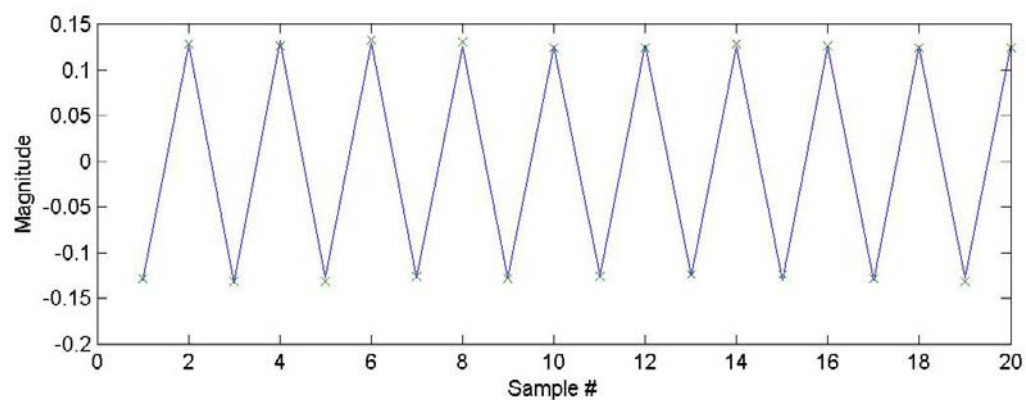
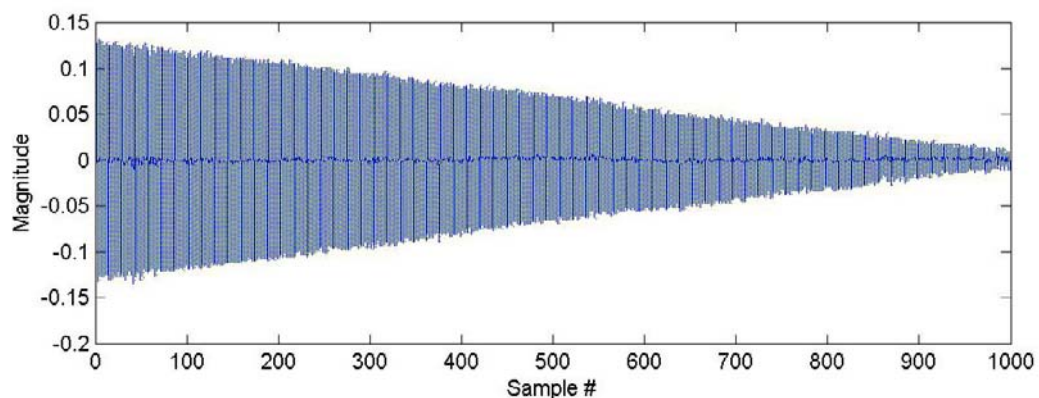
## A.10. 50.000 MHz Tone

**Description:** Test of high frequency tone reconstruction, 50 MHz.

**Input signal power spectrum:**

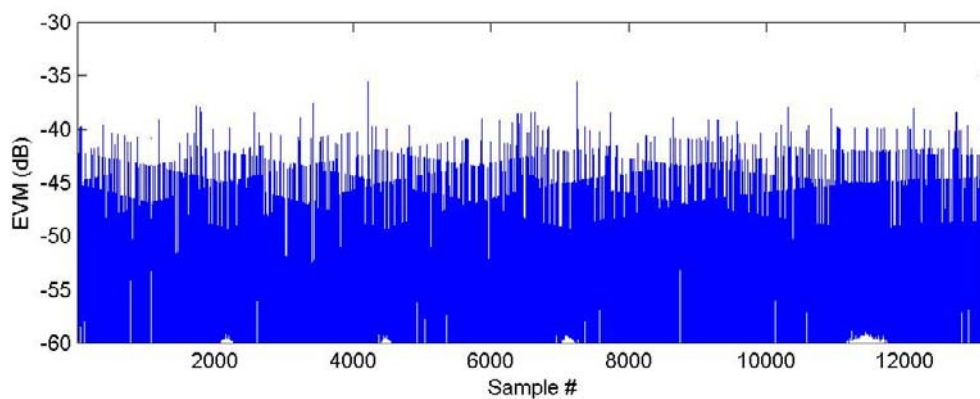


**Typical time domain input vs output:**



**Description:** Test of high frequency tone reconstruction, 50 MHz.

**Typical EVM:**

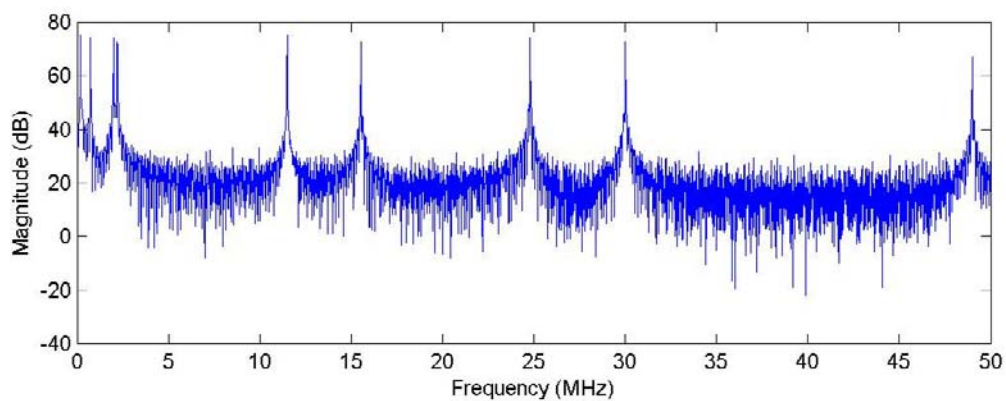


**Peak EVM:** -34.6dB

### A.11. Multi-tone Case#001

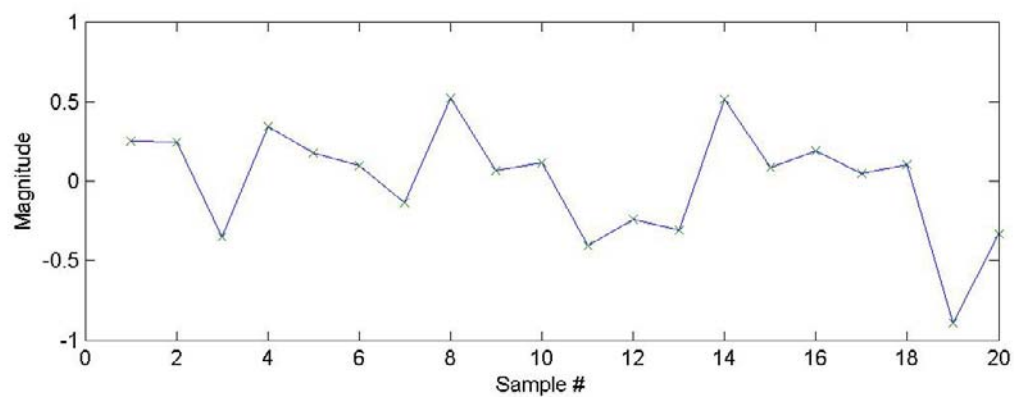
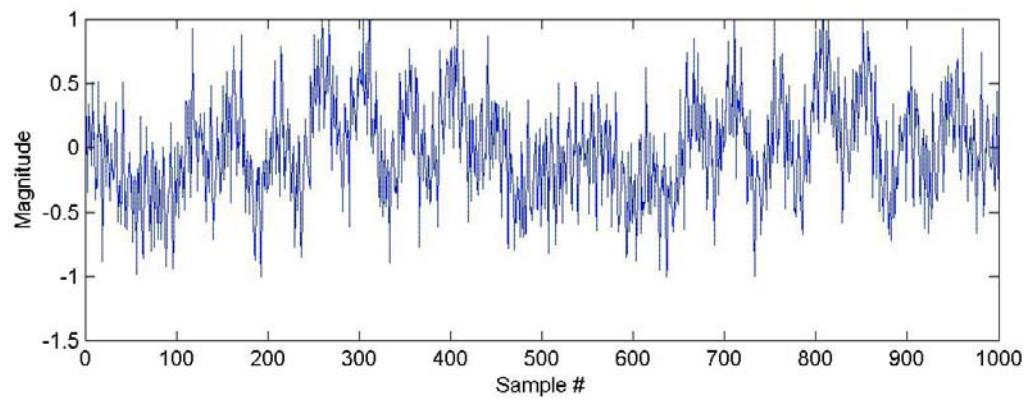
**Description:** Test of multi-tone signal reconstruction, case 1 of 4.

**Input signal power spectrum:**

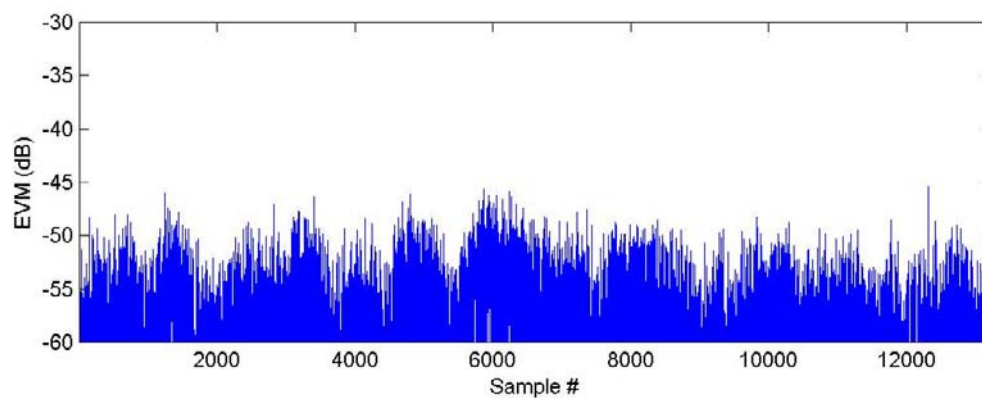


**Description:** Test of multi-tone signal reconstruction, case 1 of 4.

**Typical time domain input vs output:**



**Typical EVM:**

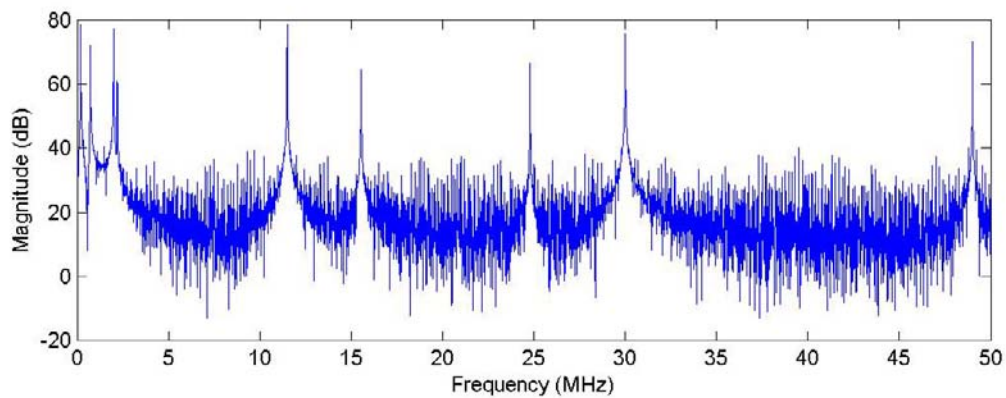


**Peak EVM:** -45.3dB

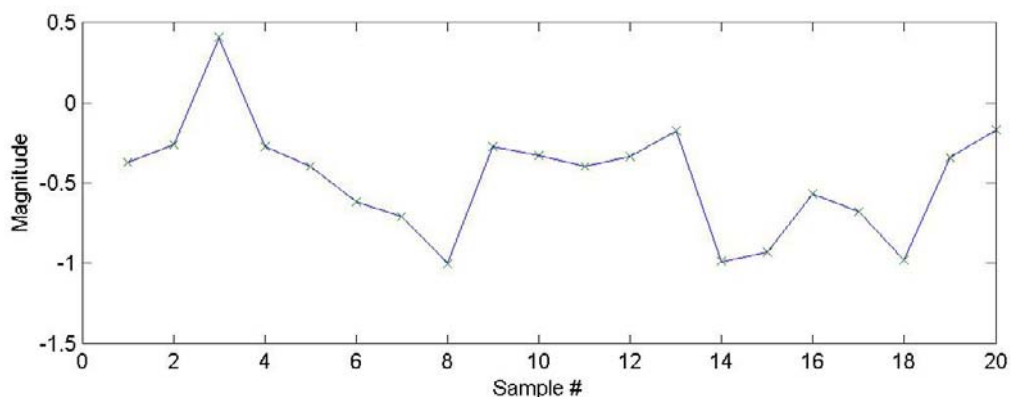
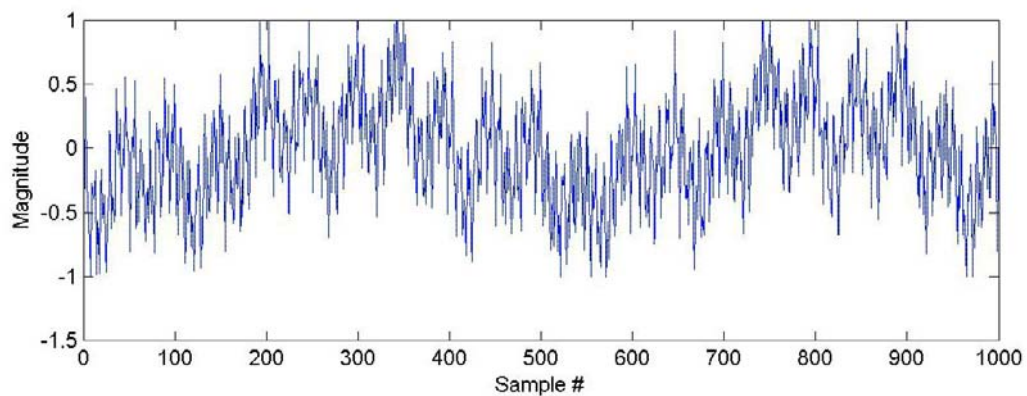
**A.12. Multi-tone Case#002**

**Description:** Test of multi-tone signal reconstruction, case 2 of 4.

**Input signal power spectrum:**



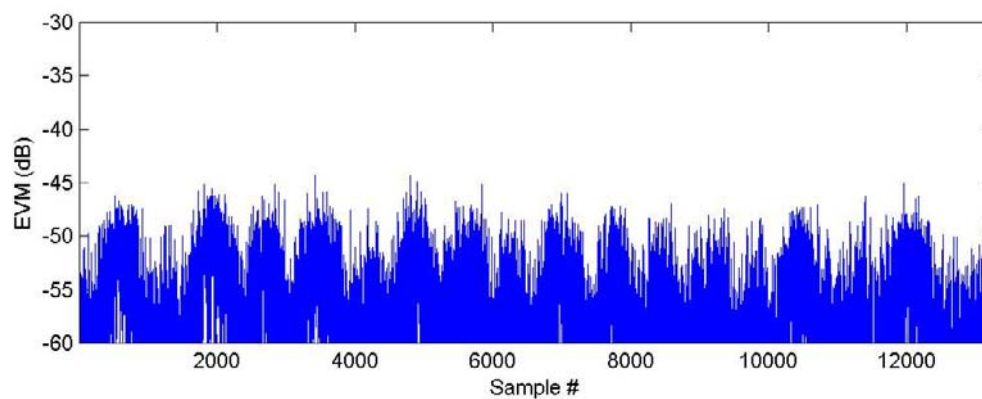
**Typical time domain input vs output:**





**Description:** Test of multi-tone signal reconstruction, case 2 of 4.

**Typical EVM:**

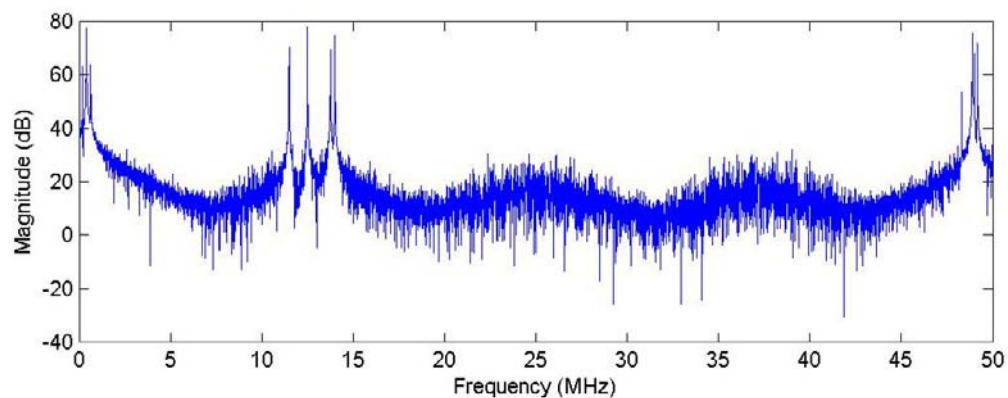


**Peak EVM:** -42.2dB

### A.13. Multi-tone Case#003

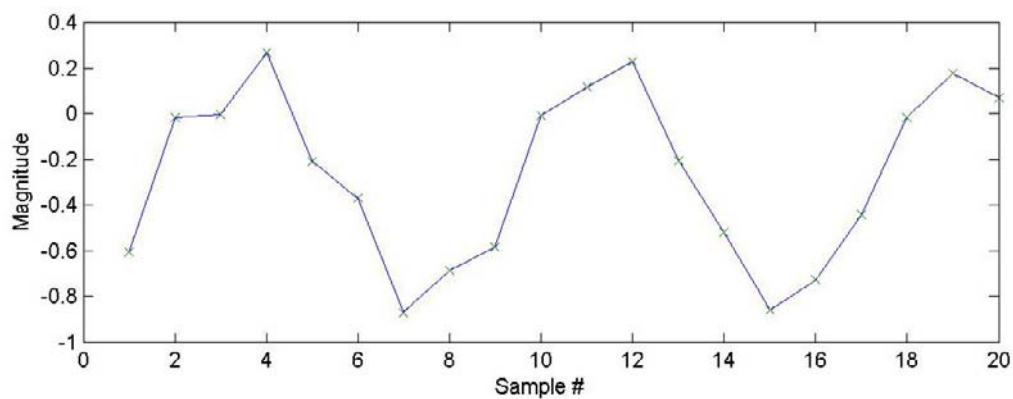
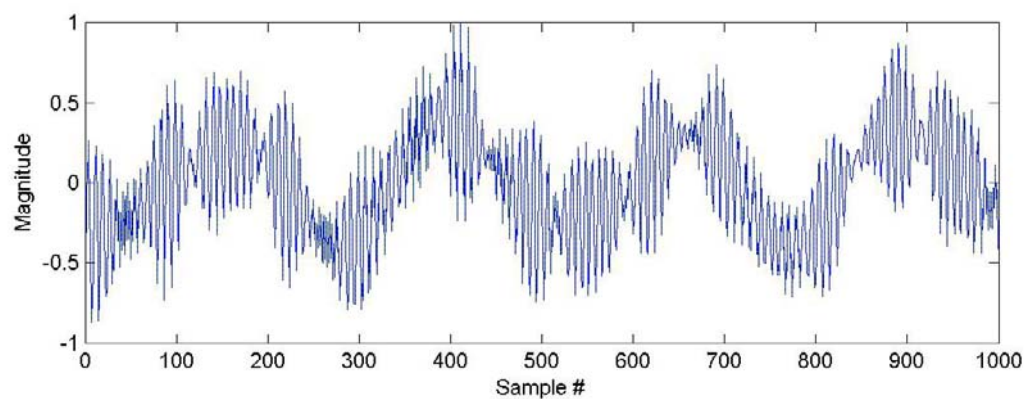
**Description:** Test of multi-tone signal reconstruction, case 3 of 4.

**Input signal power spectrum:**

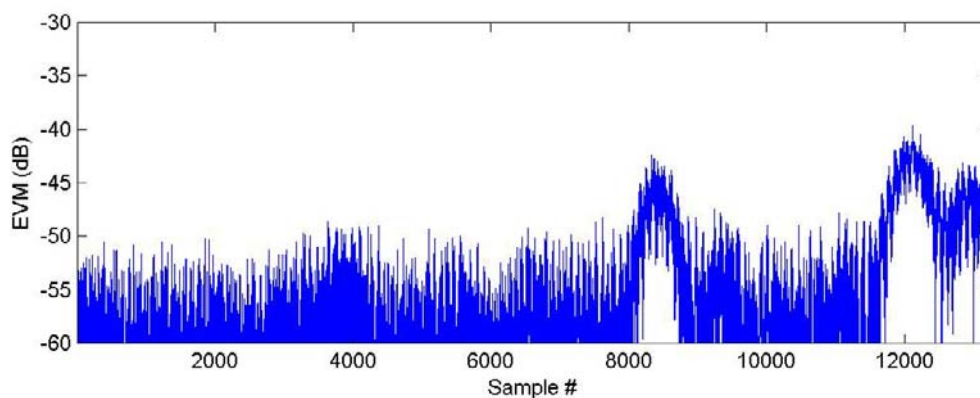


**Description:** Test of multi-tone signal reconstruction, case 3 of 4.

**Typical time domain input vs output:**



**Typical EVM:**



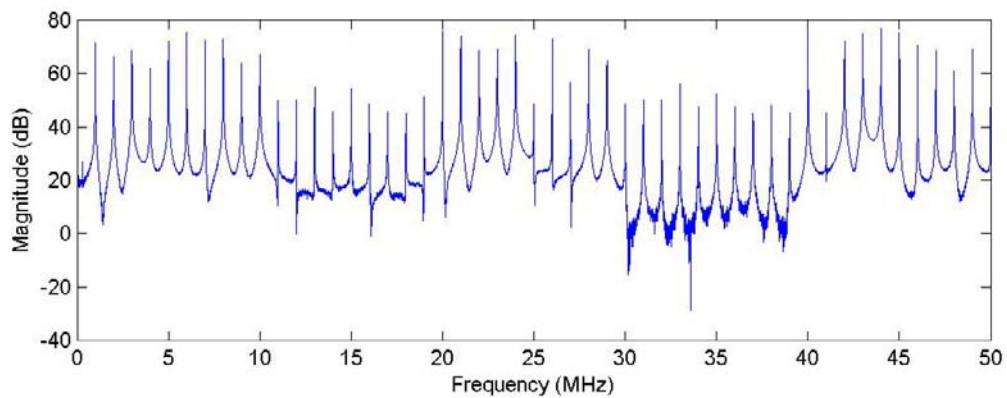
**Peak EVM:** -39.7dB



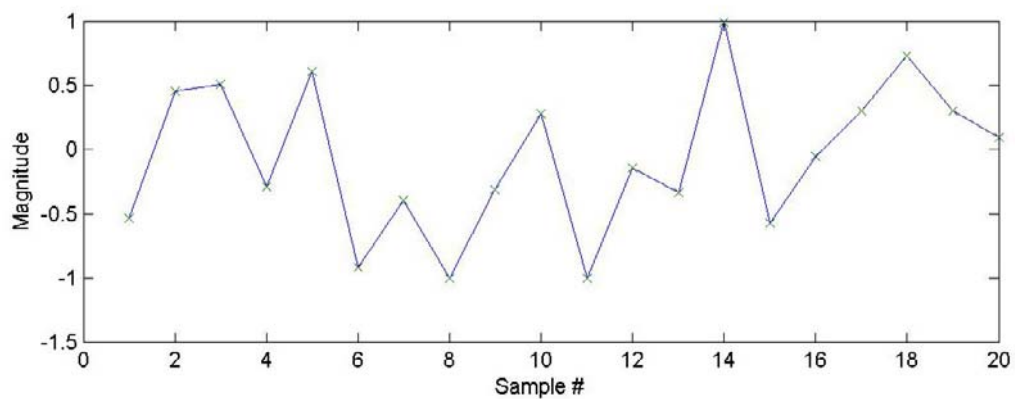
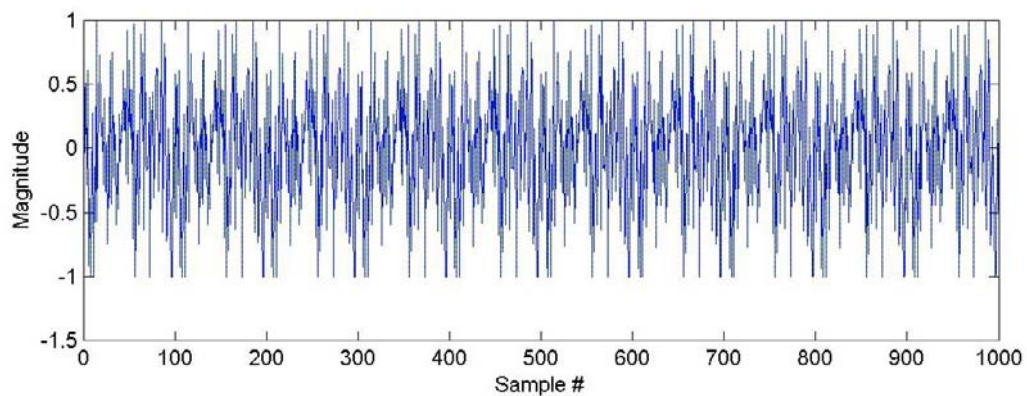
**A.14. Multi-tone Case#004**

**Description:** Test of multi-tone signal reconstruction, case 4 of 4.

**Input signal power spectrum:**

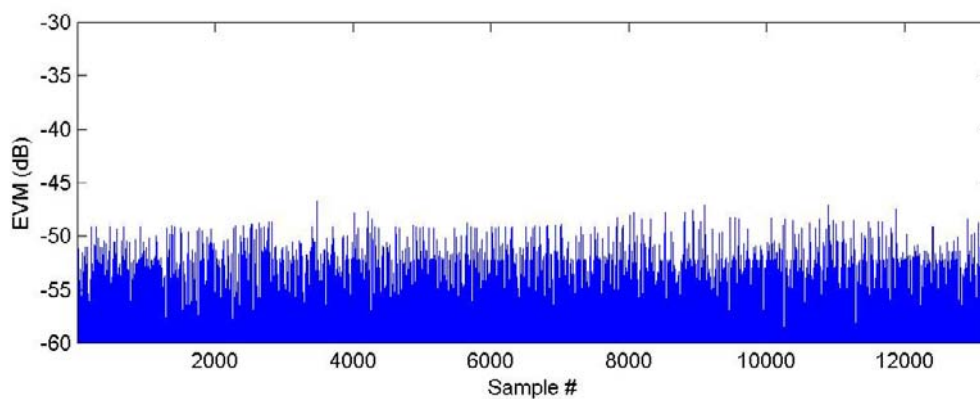


**Typical time domain input vs output:**



**Description:** Test of multi-tone signal reconstruction, case 4 of 4.

**Typical EVM:**

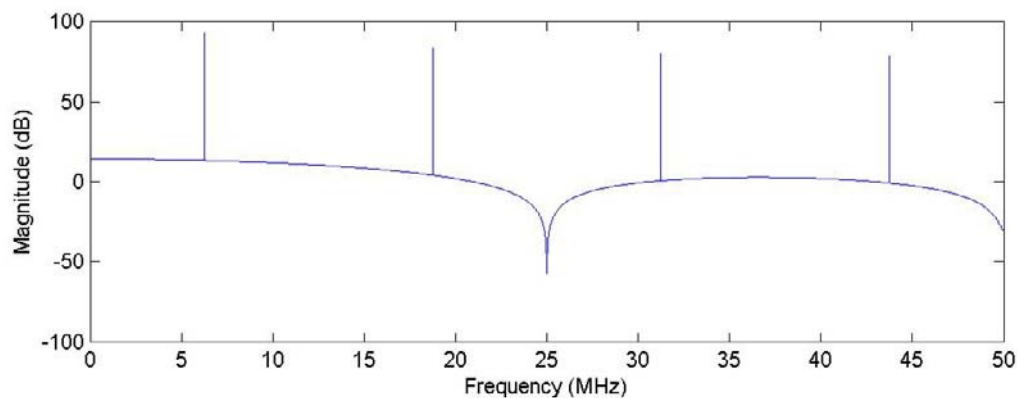


**Peak EVM:** -46.0dB

### A.15. Stored Waveform – Square Wave

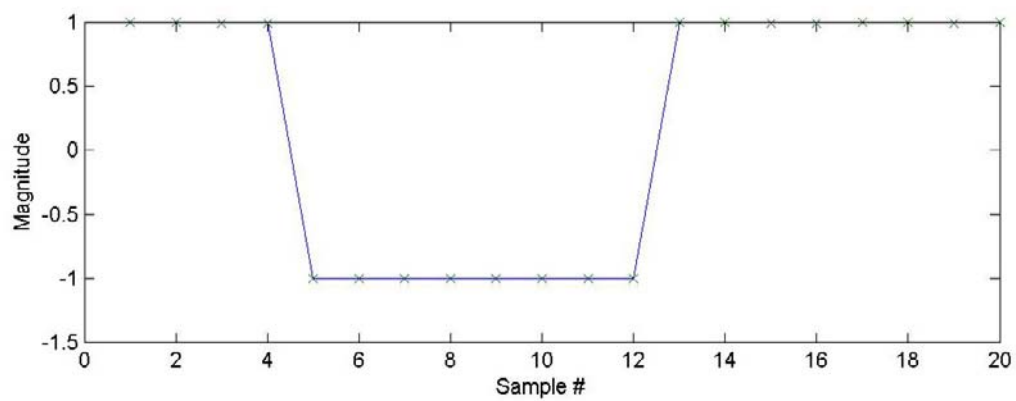
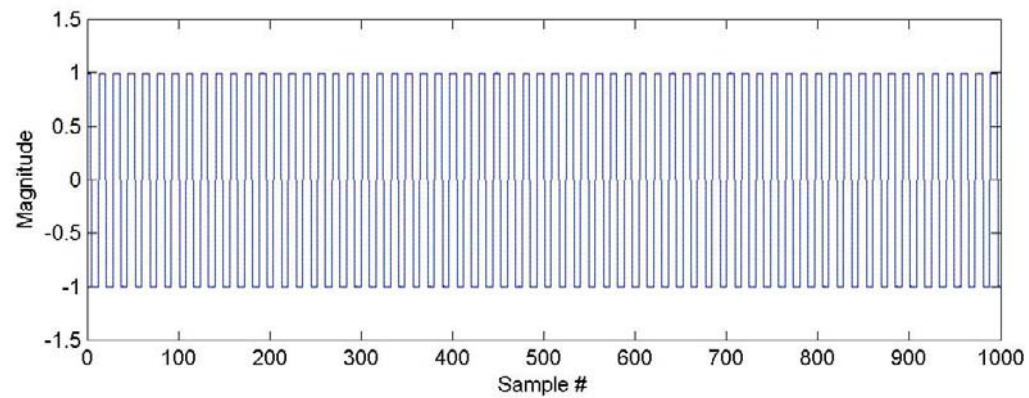
**Description:** Test of broadband signal reconstruction, square wave.

**Input signal power spectrum:**

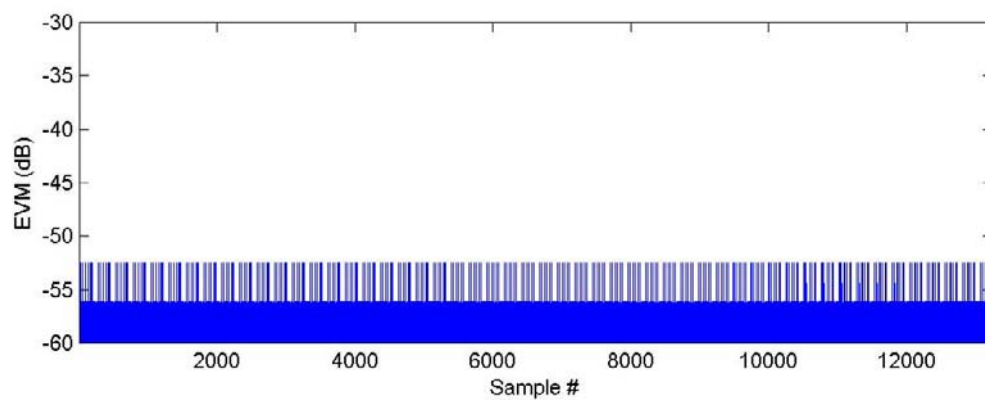


**Description:** Test of broadband signal reconstruction, square wave.

**Typical time domain input vs output:**



**Typical EVM:**

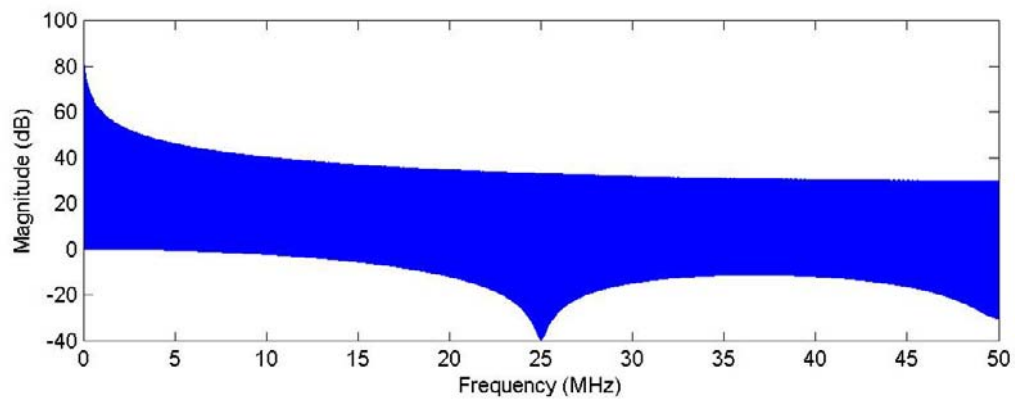


**Peak EVM:** -52.3dB

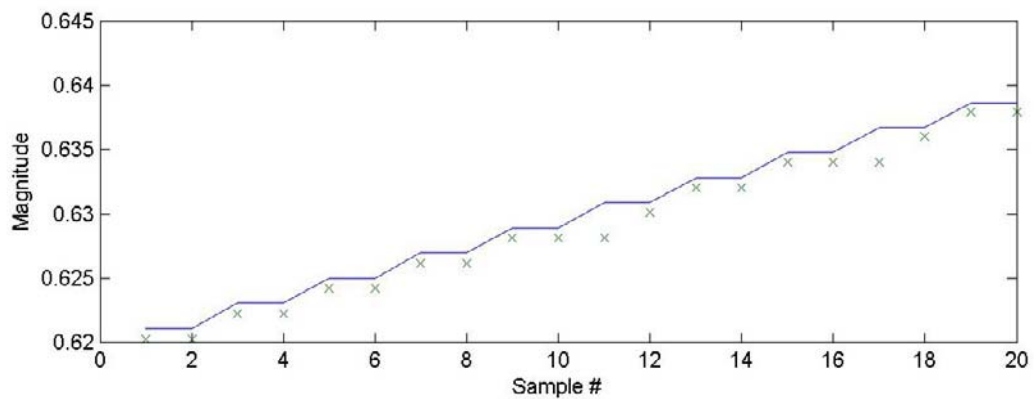
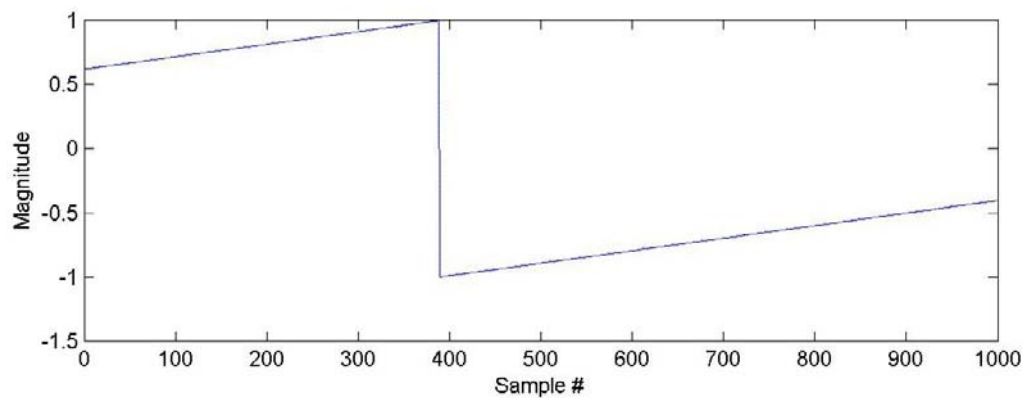
## A.16. Stored Waveform - Ramp

**Description:** Test of broadband signal reconstruction, ramp.

**Input signal power spectrum:**

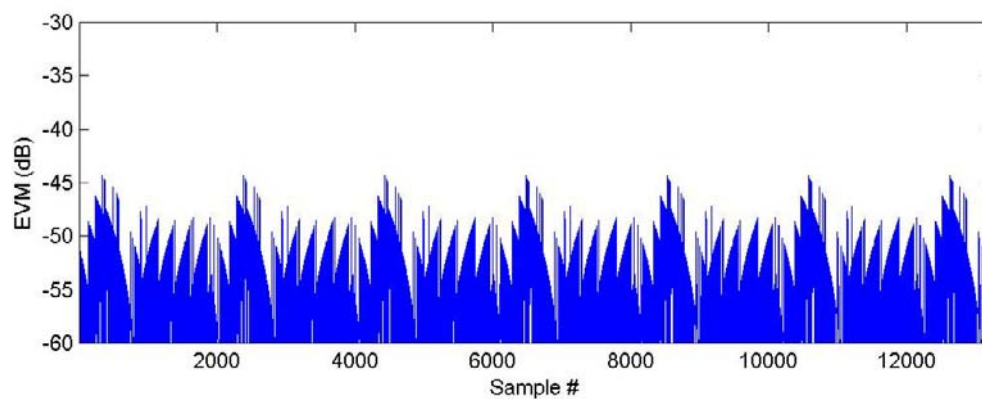


**Typical time domain input vs output:**



**Description:** Test of broadband signal reconstruction, ramp.

**Typical EVM:**

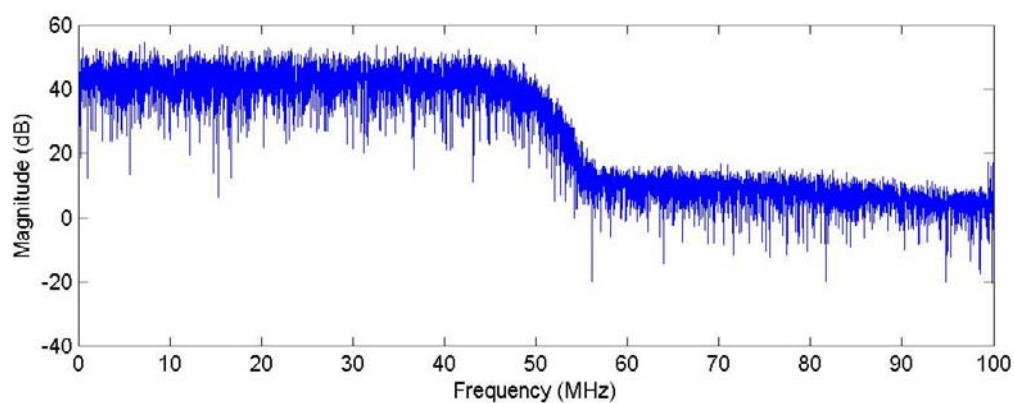


**Peak EVM:** -52.3dB

### A.17. AWGN, 200MSPS

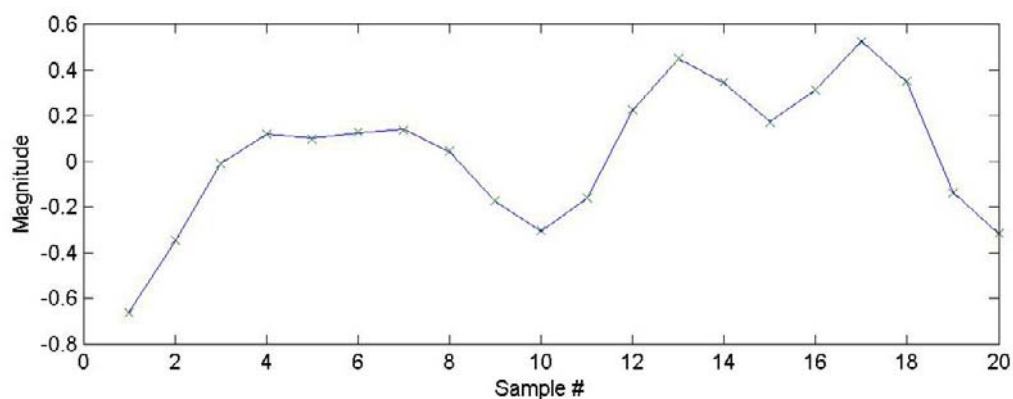
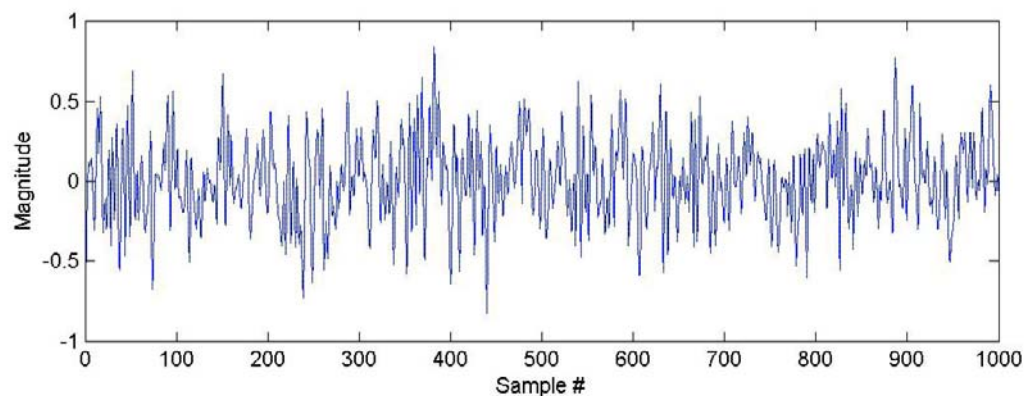
**Description:** Test of broadband signal reconstruction, COTS AWGN generator, full sample rate.

**Input signal power spectrum:**

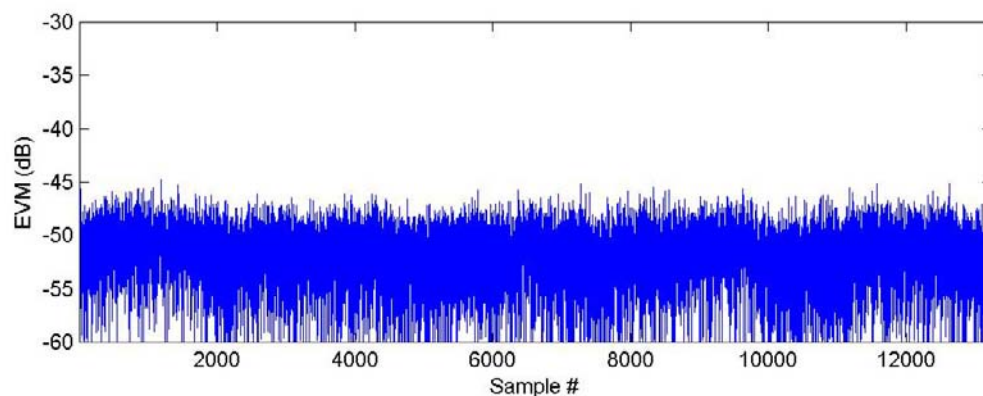


**Description:** Test of broadband signal reconstruction, COTS AWGN generator, full sample rate.

**Typical time domain input vs output:**



**Typical EVM:**



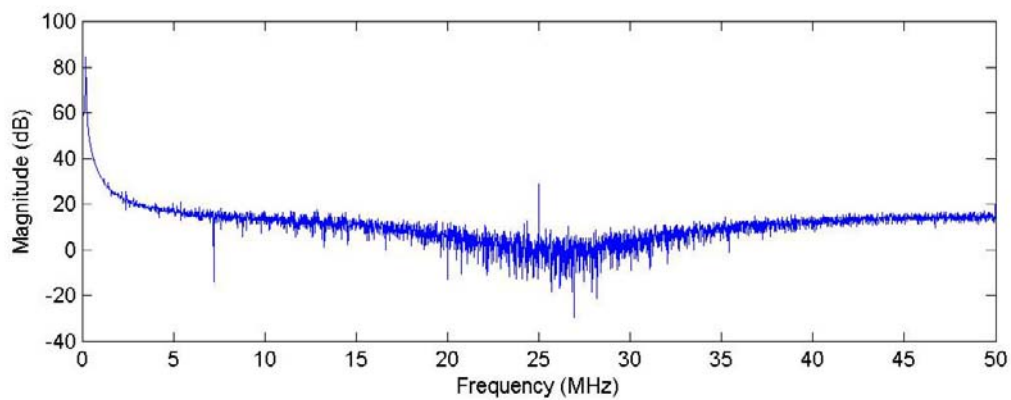
**Peak EVM:** -43.2dB

## Appendix B: 4096-bin Filterbank Acceptance Test Results

### B.1. 200 kHz Tone

**Description:** Test of low-frequency tone reconstruction, 200 kHz.

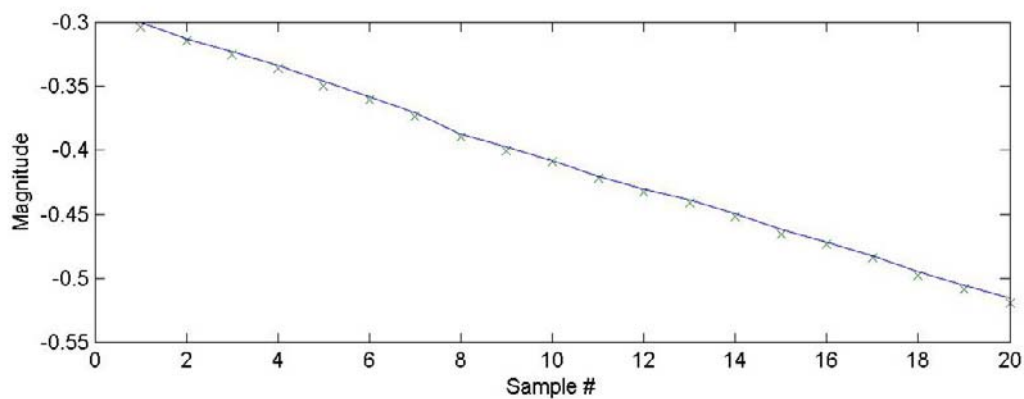
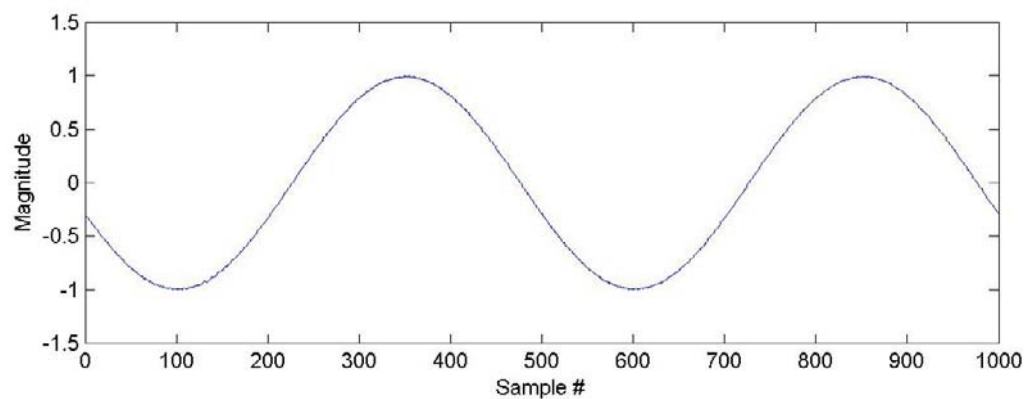
**Input signal power spectrum:**



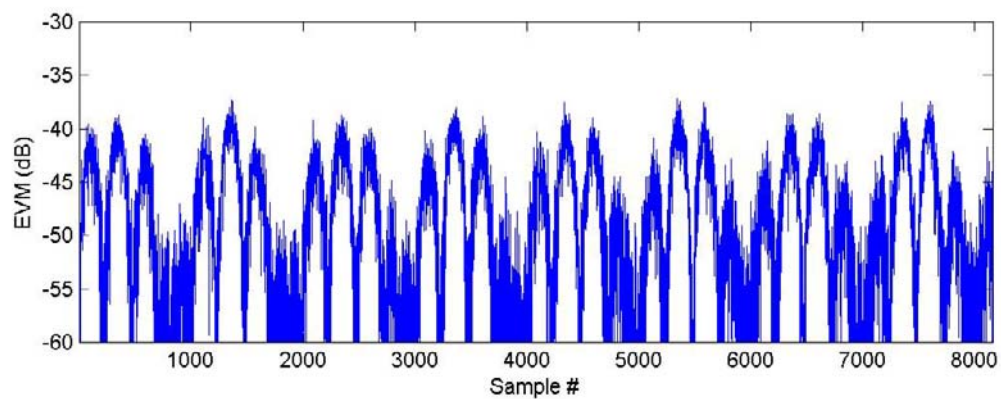


**Description:** Test of low-frequency tone reconstruction, 200 kHz.

**Typical time domain input vs output:**



**Typical EVM:**



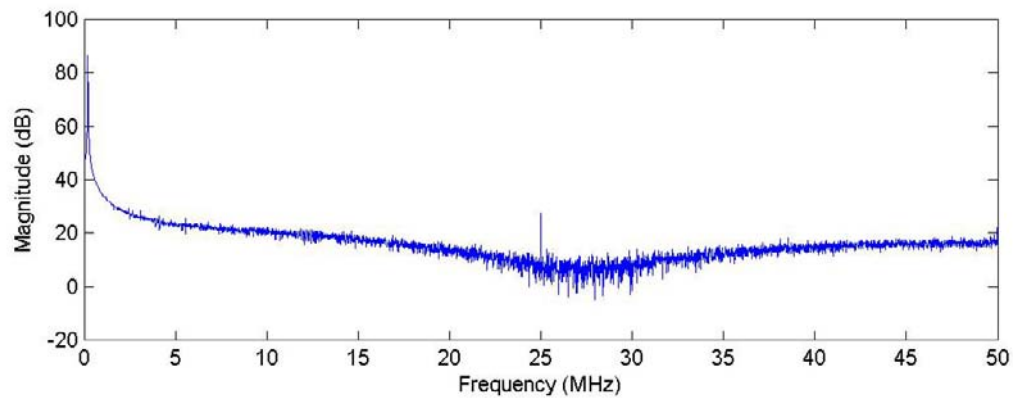
**Peak EVM:** -37.1dB



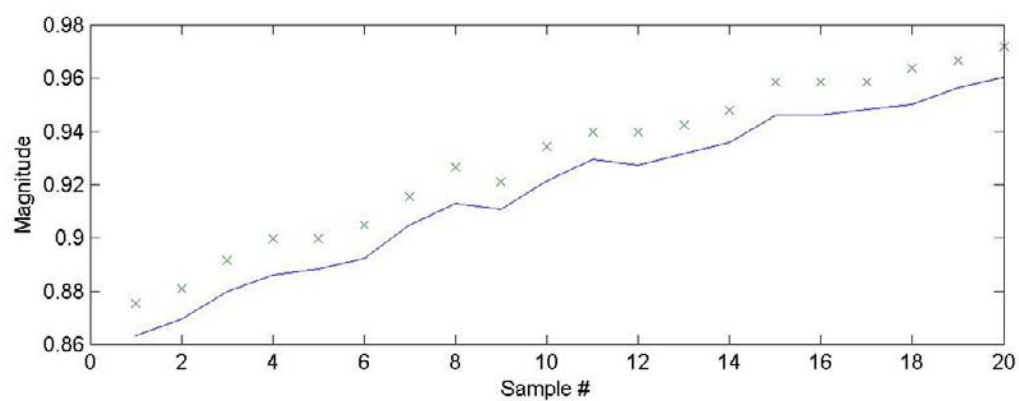
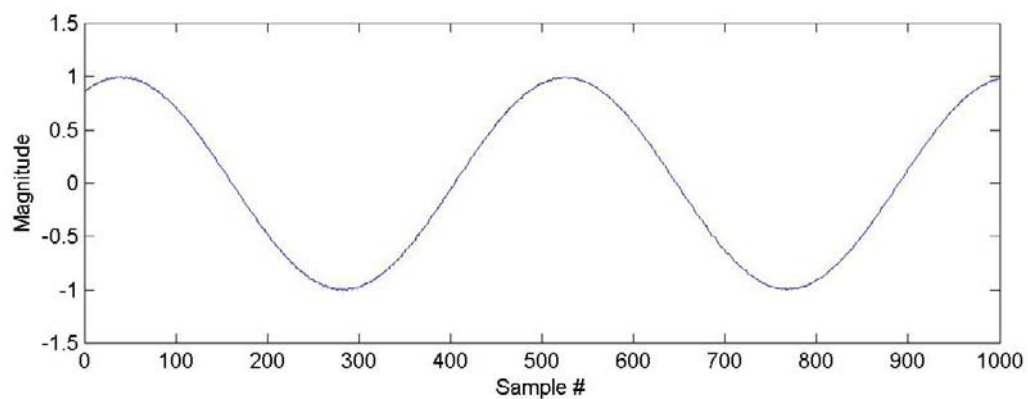
## B.2. 206 kHz Tone

**Description:** Test of low-frequency tone reconstruction, 200 kHz +  $\frac{1}{4}$ -bin.

**Input signal power spectrum:**

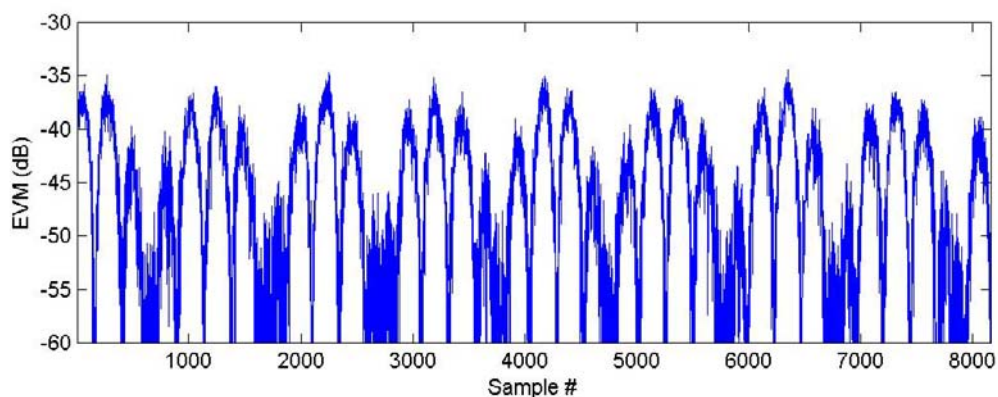


**Typical time domain input vs output:**



**Description:** Test of low-frequency tone reconstruction, 200 kHz +  $\frac{1}{4}$ -bin.

**Typical EVM:**

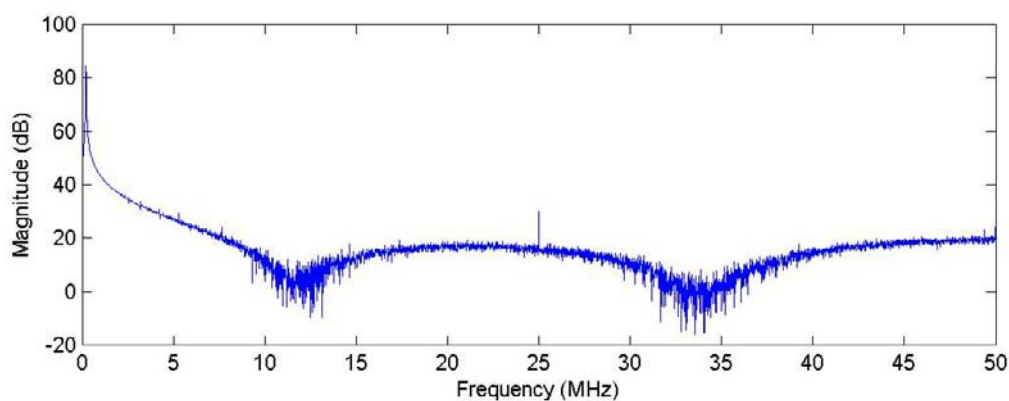


**Peak EVM:** -34.4dB

### B.3. 212 kHz Tone

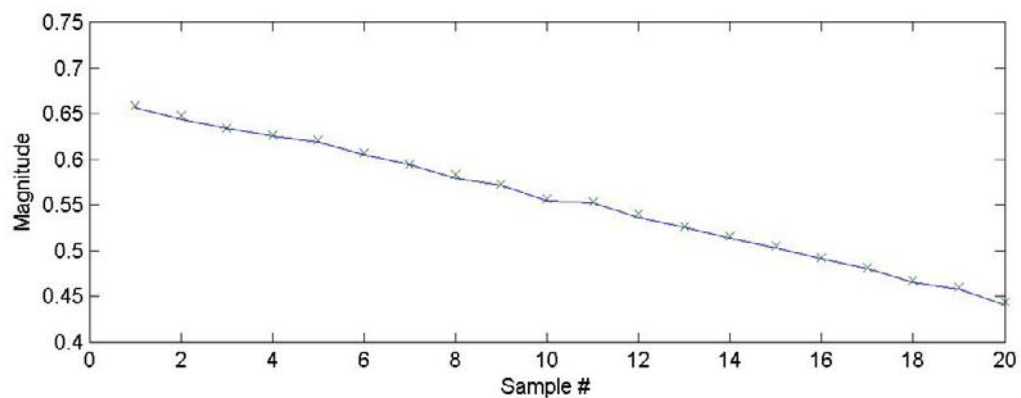
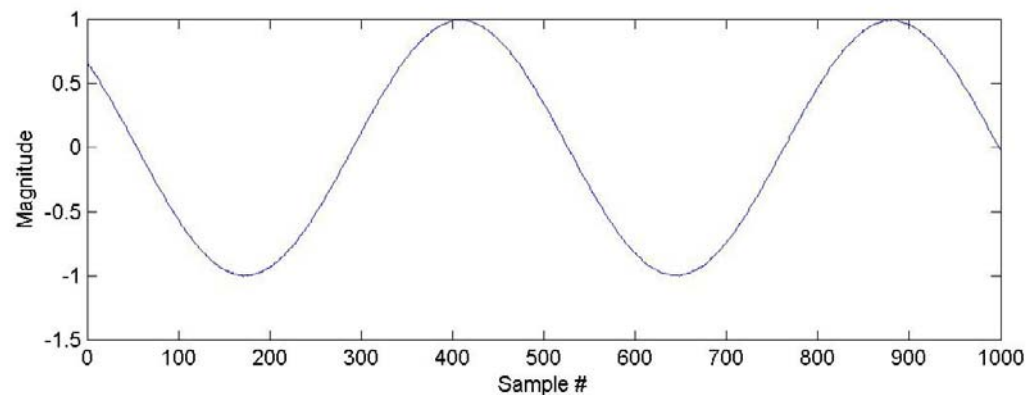
**Description:** Test of low-frequency tone reconstruction, 200 kHz +  $\frac{1}{2}$  -bin.

**Input signal power spectrum:**

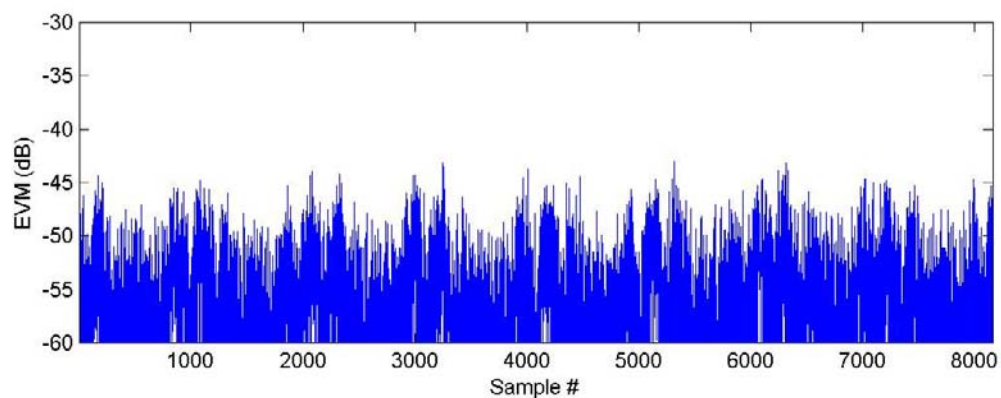


**Description:** Test of low-frequency tone reconstruction, 200 kHz +  $\frac{1}{2}$  -bin.

**Typical time domain input vs output:**



**Typical EVM:**

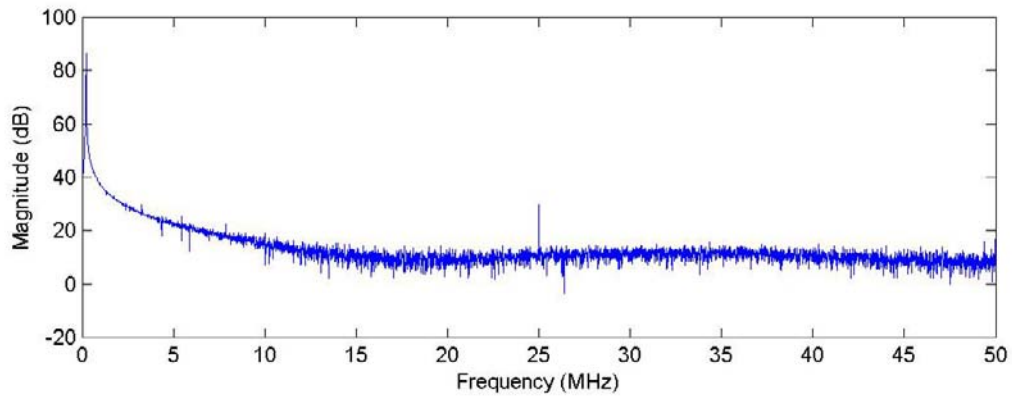


**Peak EVM:** -42.9dB

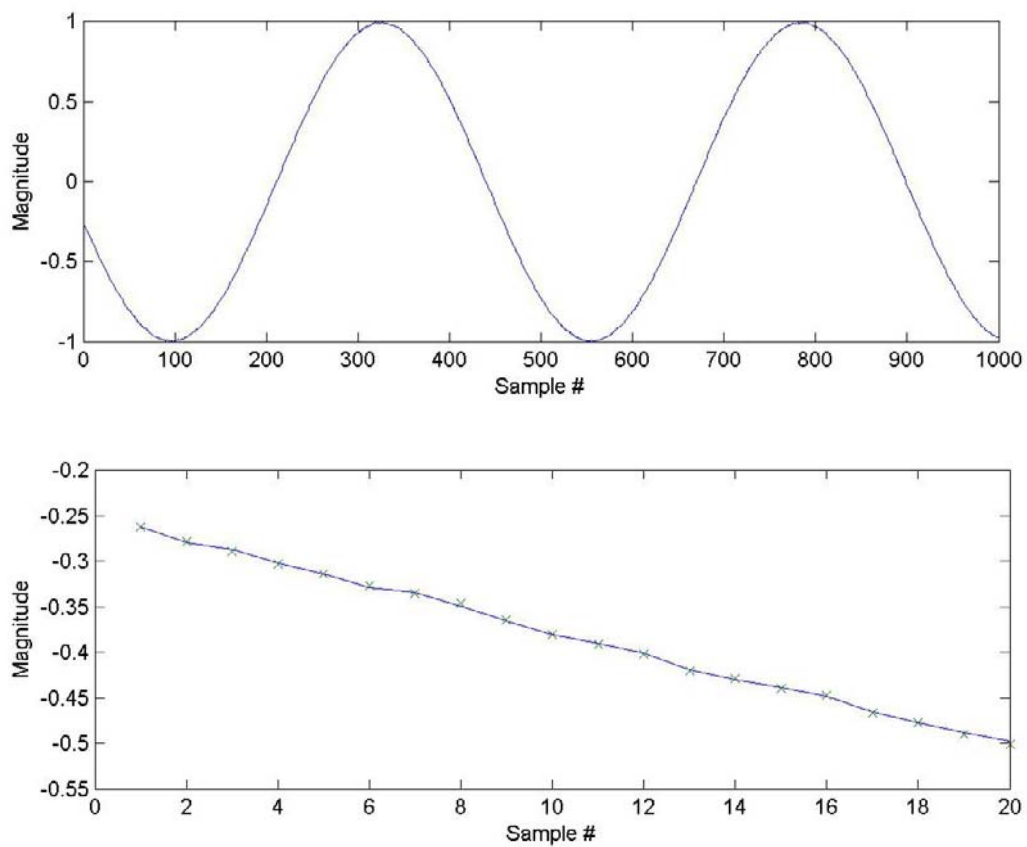
## B.4. 218 kHz Tone

**Description:** Test of low-frequency tone reconstruction, 200 kHz +  $\frac{3}{4}$  -bin.

**Input signal power spectrum:**

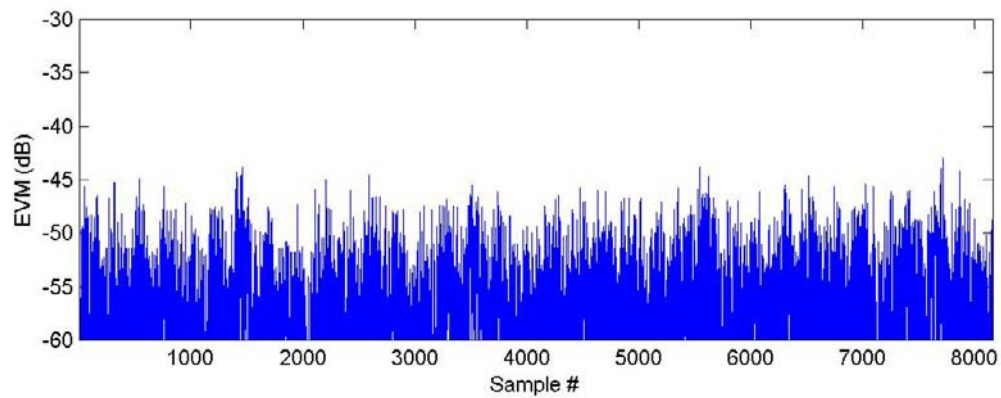


**Typical time domain input vs output:**



**Description:** Test of low-frequency tone reconstruction, 200 kHz +  $\frac{3}{4}$  -bin.

**Typical EVM:**

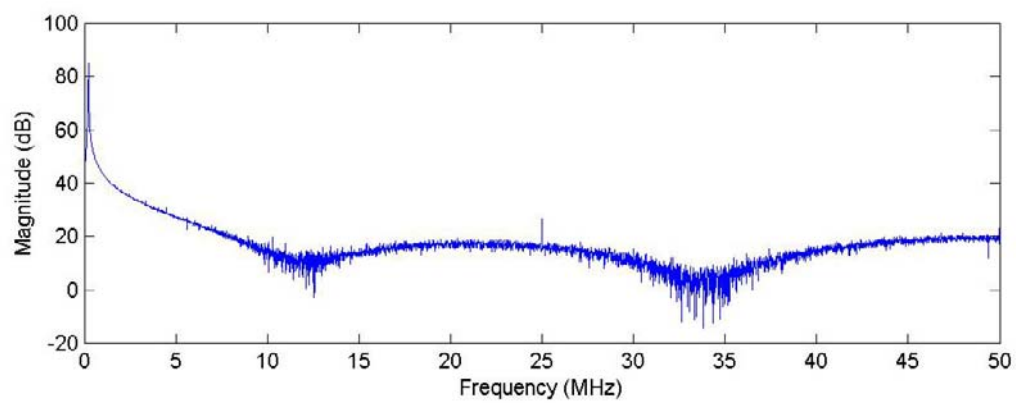


**Peak EVM:** -42.9dB

## B.5. 224 kHz Tone

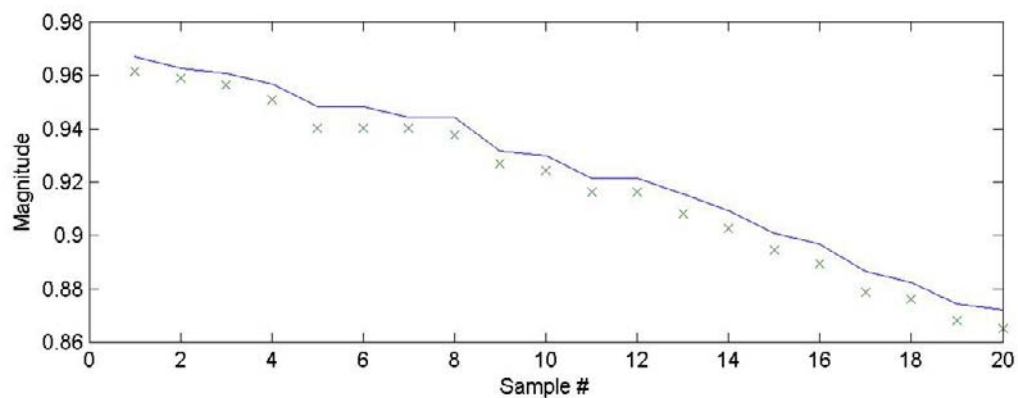
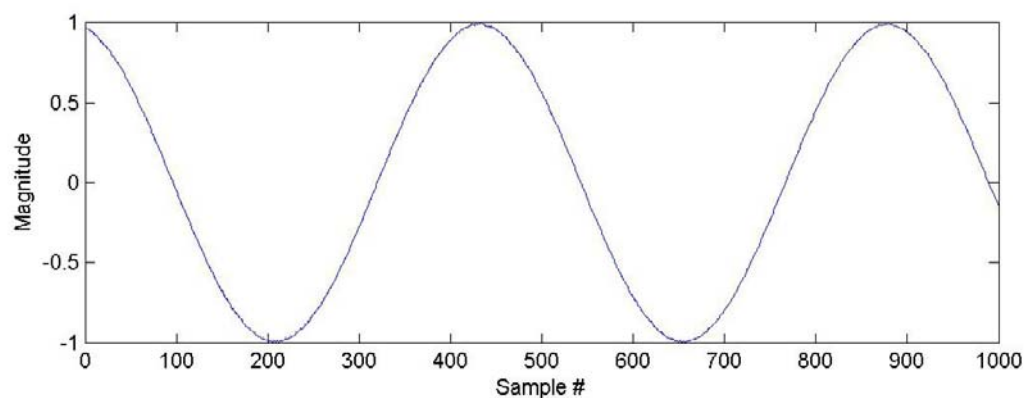
**Description:** Test of low-frequency tone reconstruction, 200 kHz + 1-bin.

**Input signal power spectrum:**

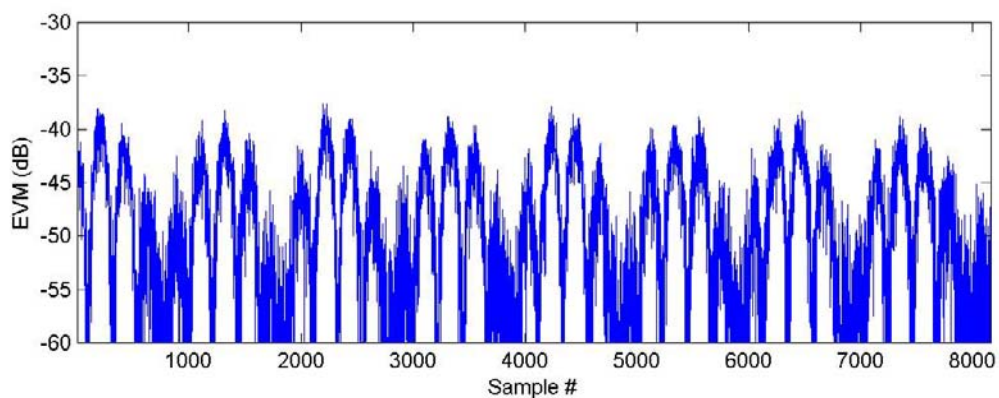


**Description:** Test of low-frequency tone reconstruction, 200 kHz + 1-bin.

**Typical time domain input vs output:**



**Typical EVM:**

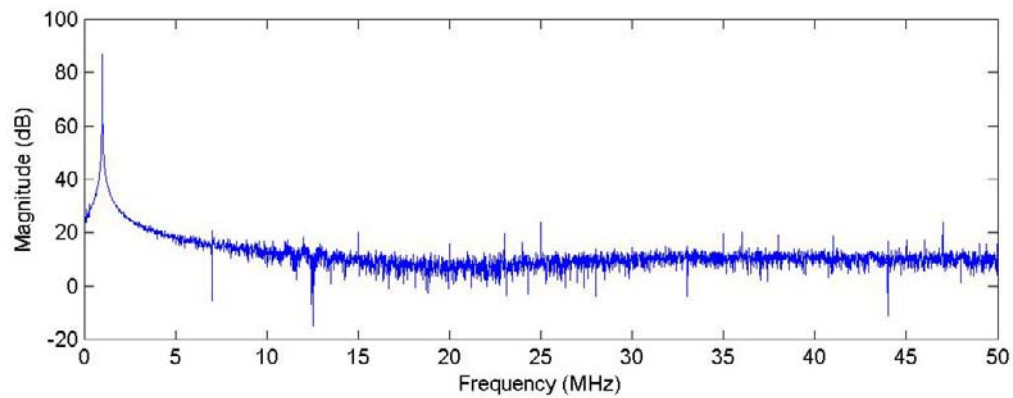


**Peak EVM:** -37.6dB

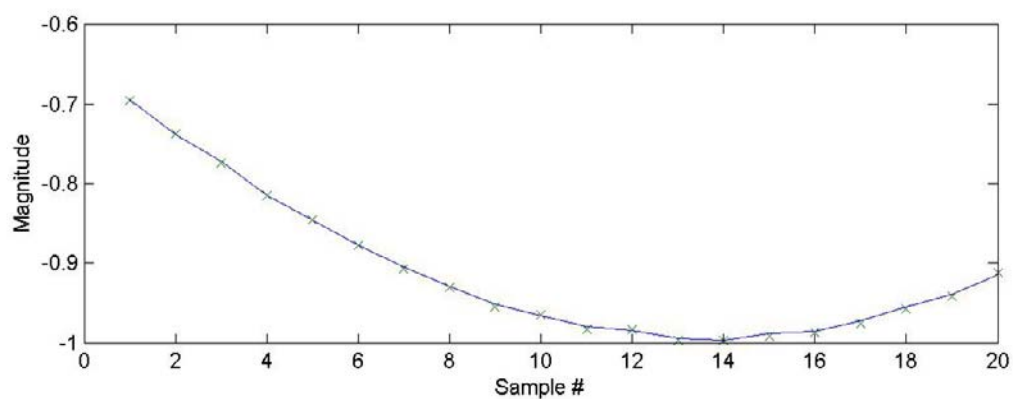
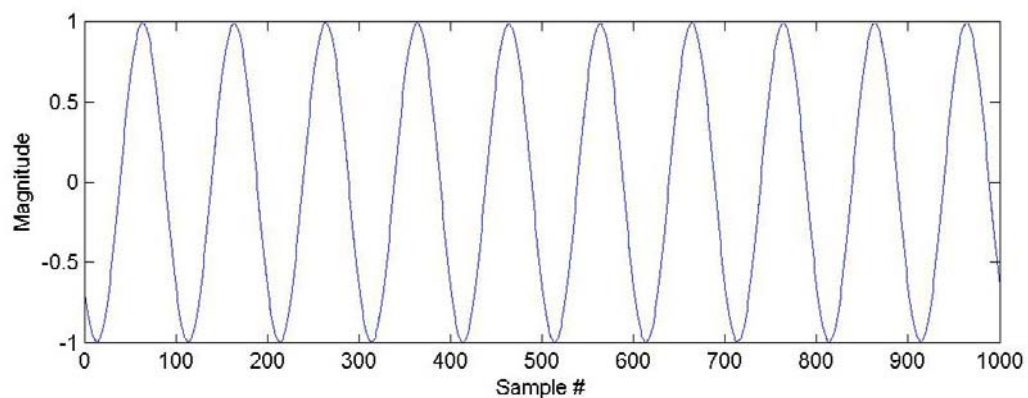
## B.6. 1 MHz Tone

**Description:** Test of low-frequency tone reconstruction, 1 MHz.

**Input signal power spectrum:**

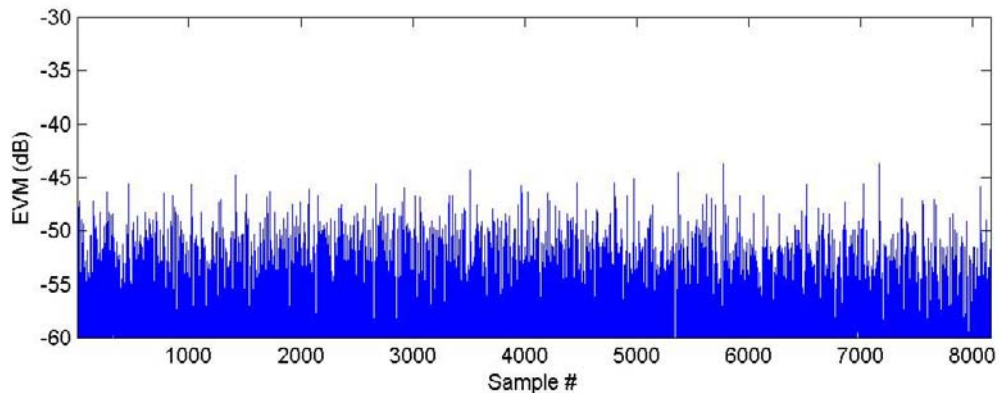


**Typical time domain input vs output:**



**Description:** Test of low-frequency tone reconstruction, 1 MHz.

**Typical EVM:**

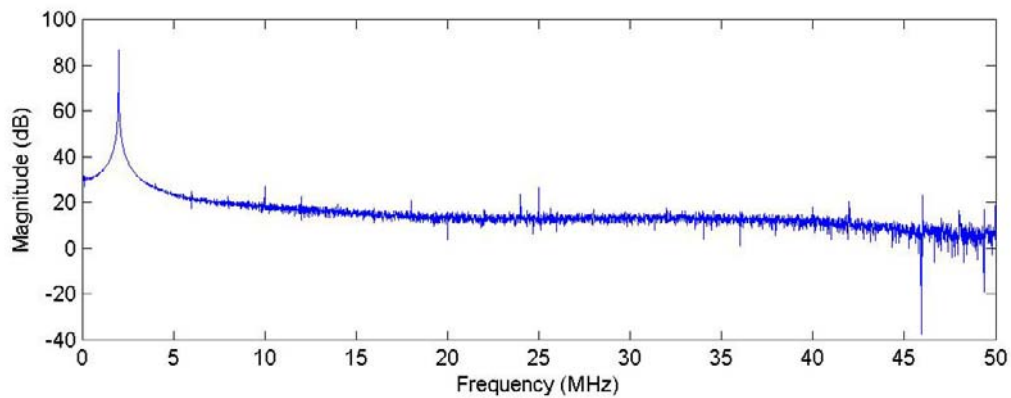


**Peak EVM:** -43.6dB

## B.7. 2 MHz Tone

**Description:** Test of low-frequency tone reconstruction, 2 MHz.

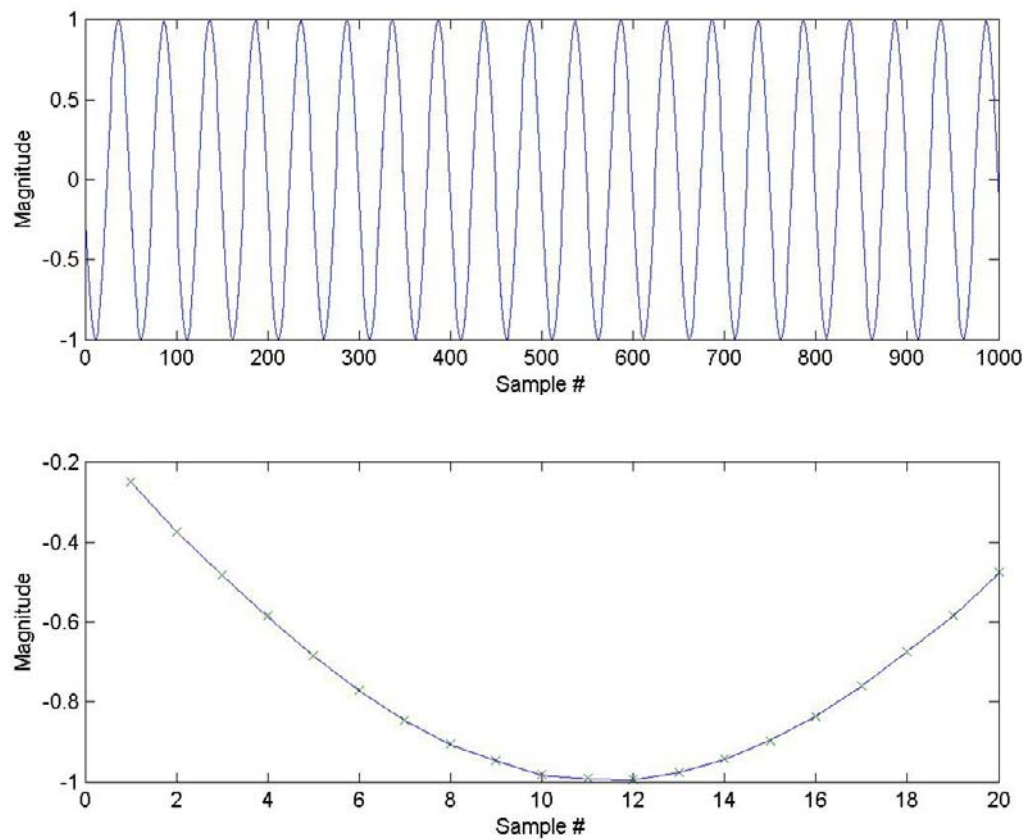
**Input signal power spectrum:**



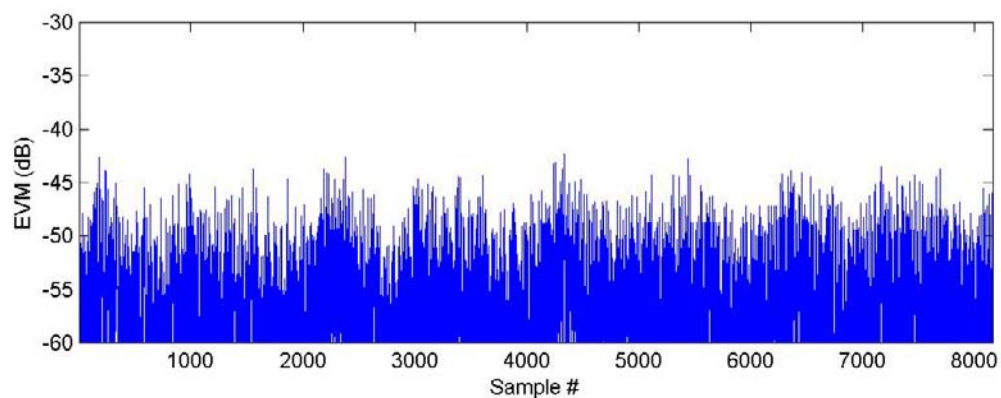


**Description:** Test of low-frequency tone reconstruction, 2 MHz.

**Typical time domain input vs output:**



**Typical EVM:**

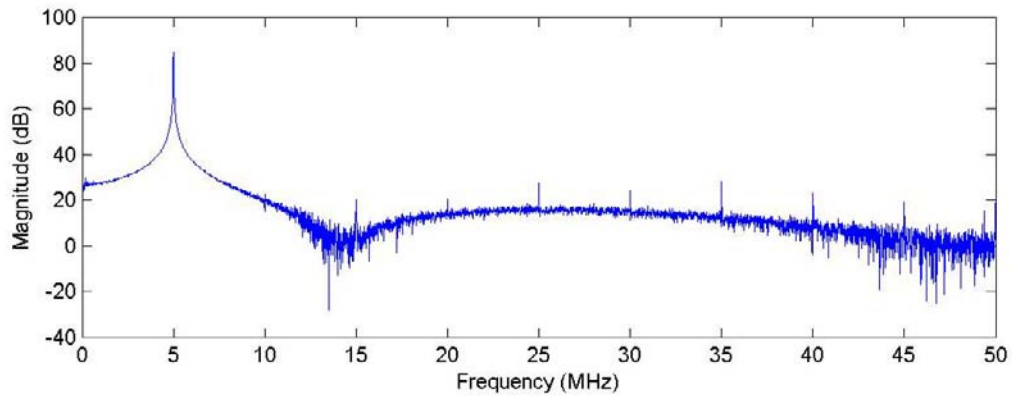


**Peak EVM:** -41.6dB

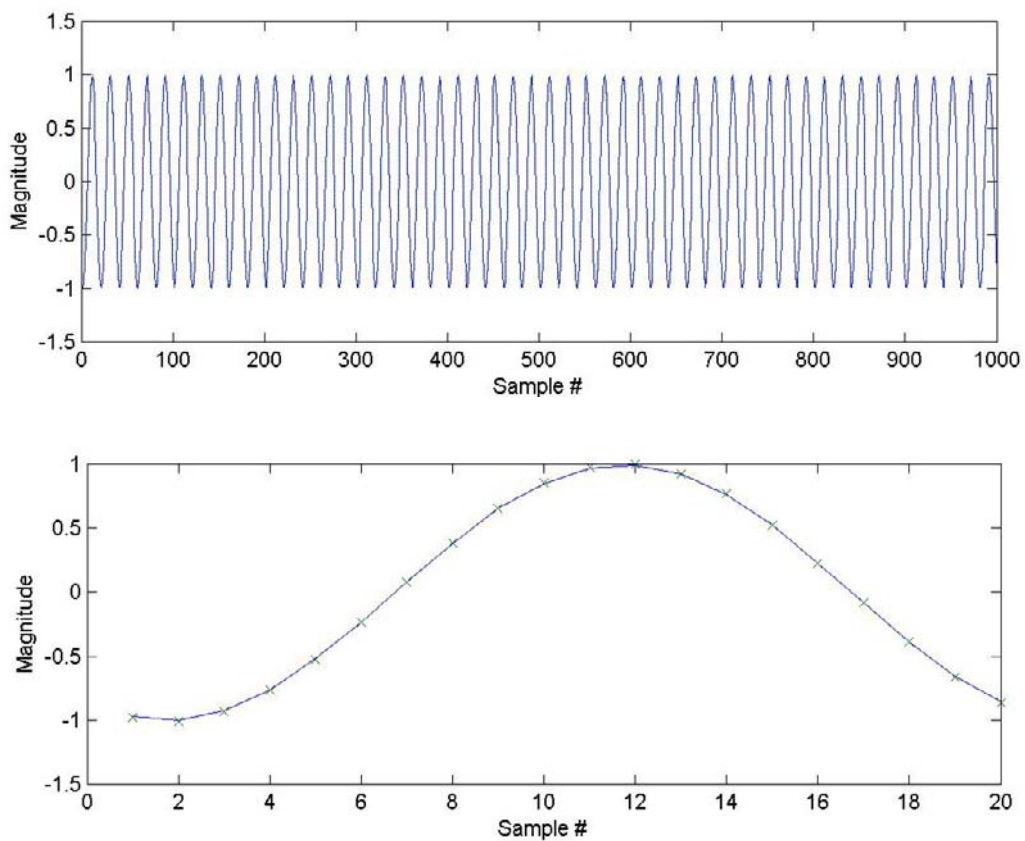
## B.8. 5 MHz Tone

**Description:** Test of low-frequency tone reconstruction, 5 MHz.

**Input signal power spectrum:**

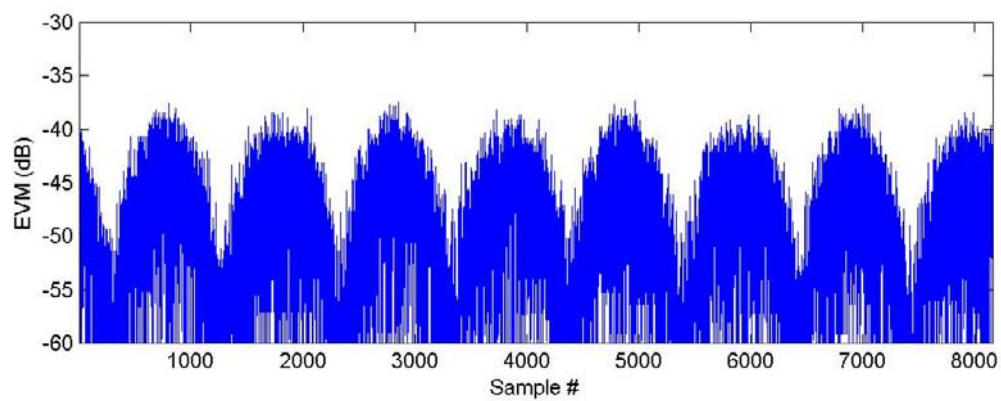


**Typical time domain input vs output:**



**Description:** Test of low-frequency tone reconstruction, 5 MHz.

**Typical EVM:**

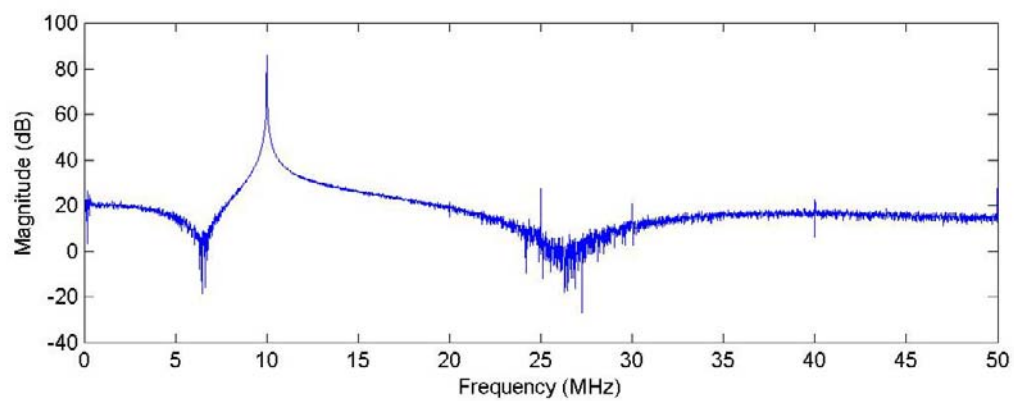


**Peak EVM:** -36.7dB

## B.9. 10 MHz Tone

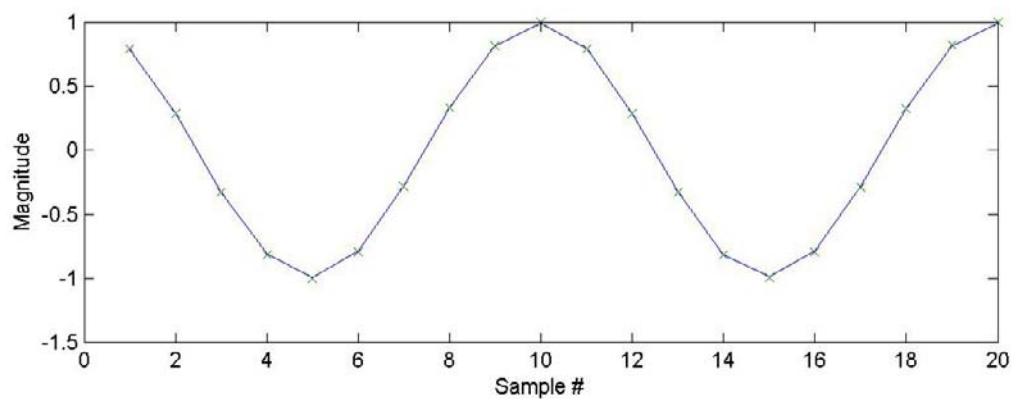
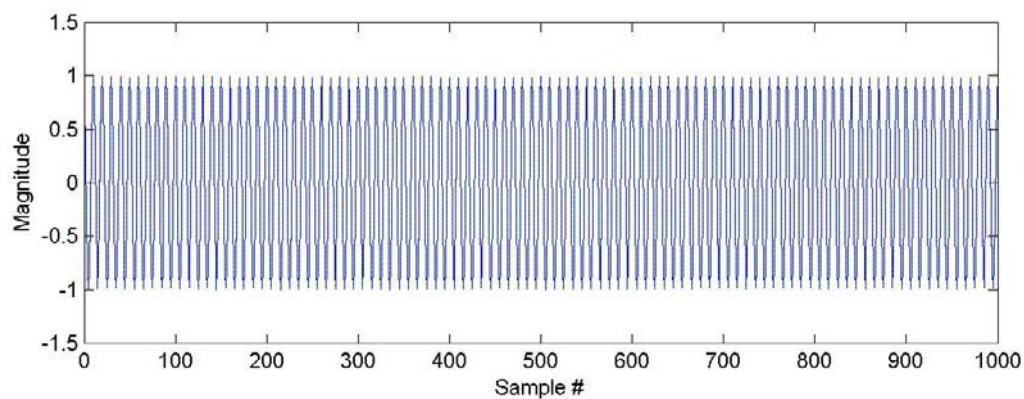
**Description:** Test of tone reconstruction, 10 MHz.

**Input signal power spectrum:**

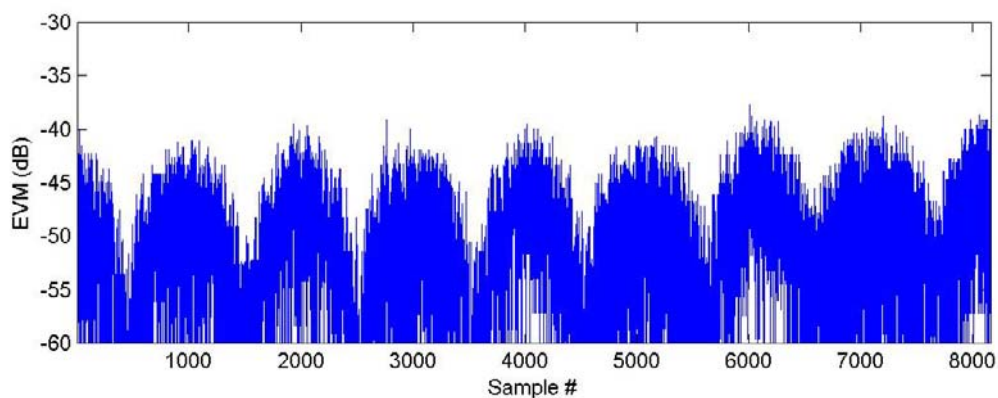


**Description:** Test of tone reconstruction, 10 MHz.

**Typical time domain input vs output:**



**Typical EVM:**

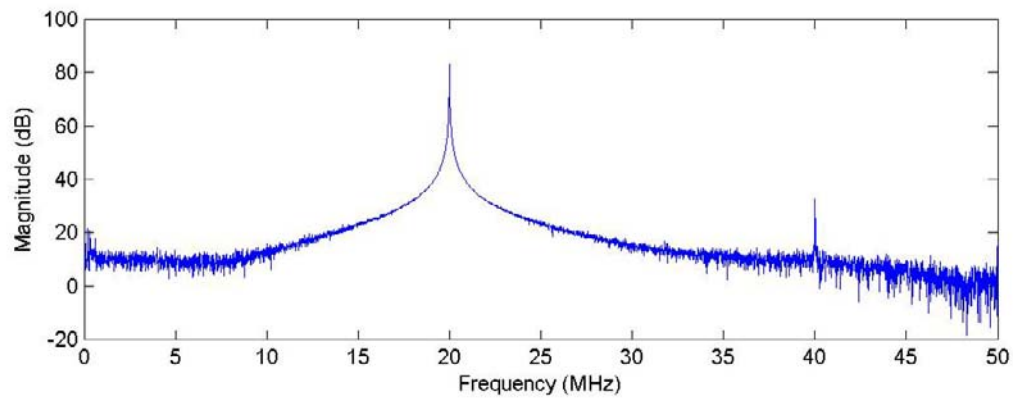


**Peak EVM:** -37.7dB

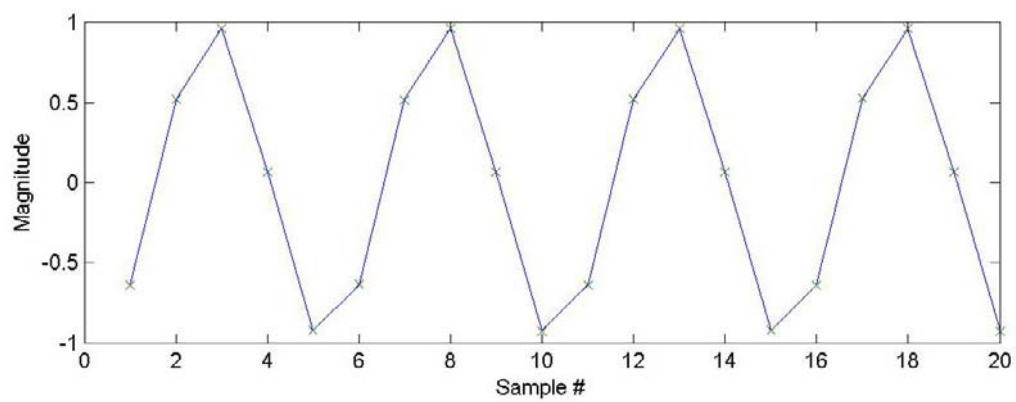
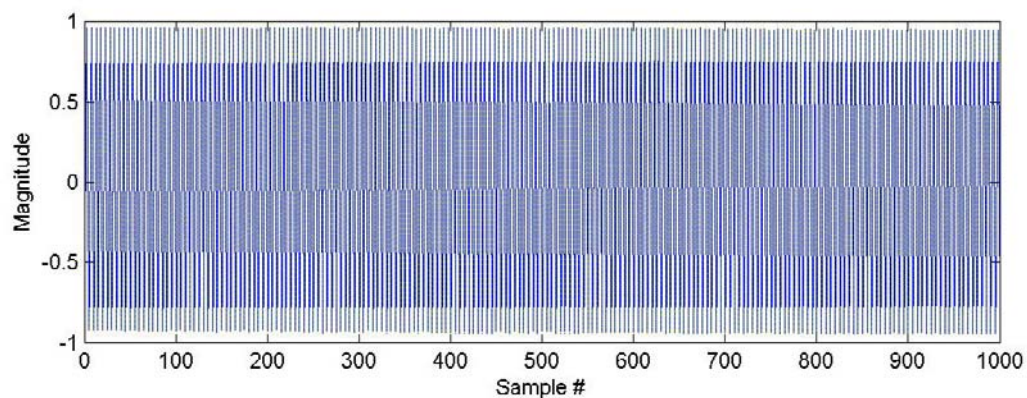
## B.10. 20 MHz Tone

**Description:** Test of tone reconstruction, 20 MHz.

**Input signal power spectrum:**

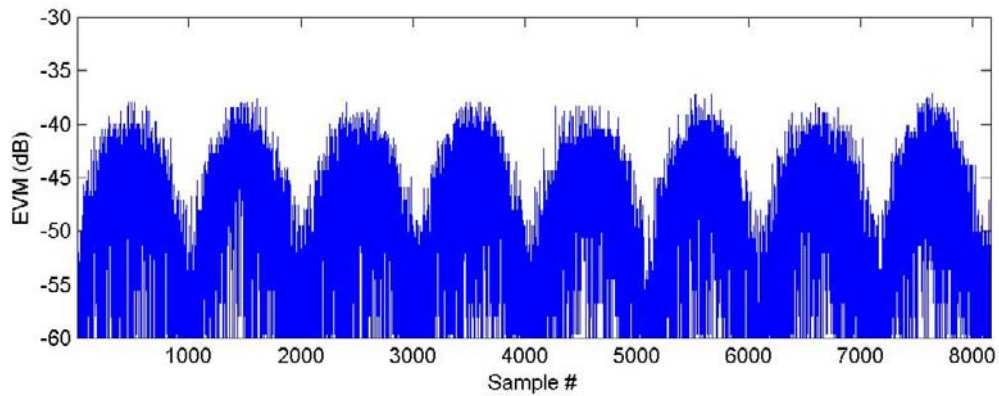


**Typical time domain input vs output:**



**Description:** Test of tone reconstruction, 20 MHz.

**Typical EVM:**

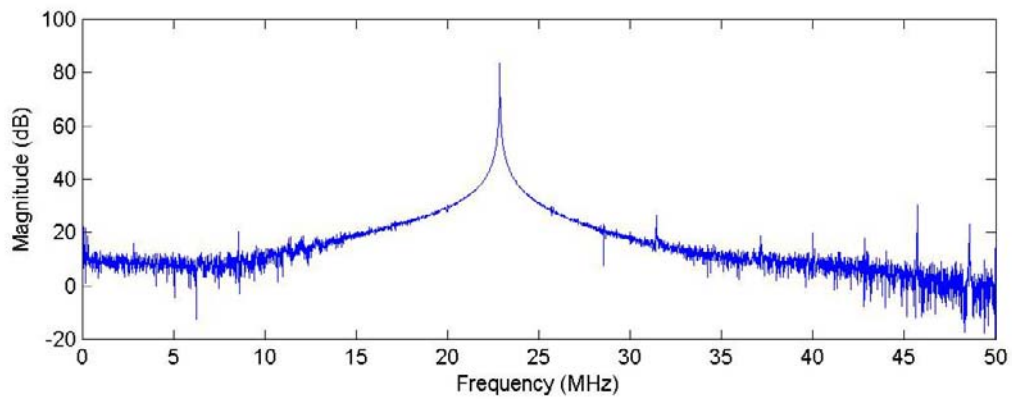


**Peak EVM:** -37.0dB

### B.11. 22.857373 MHz Tone

**Description:** Test of tone reconstruction, arbitrary frequency 22.857373 MHz.

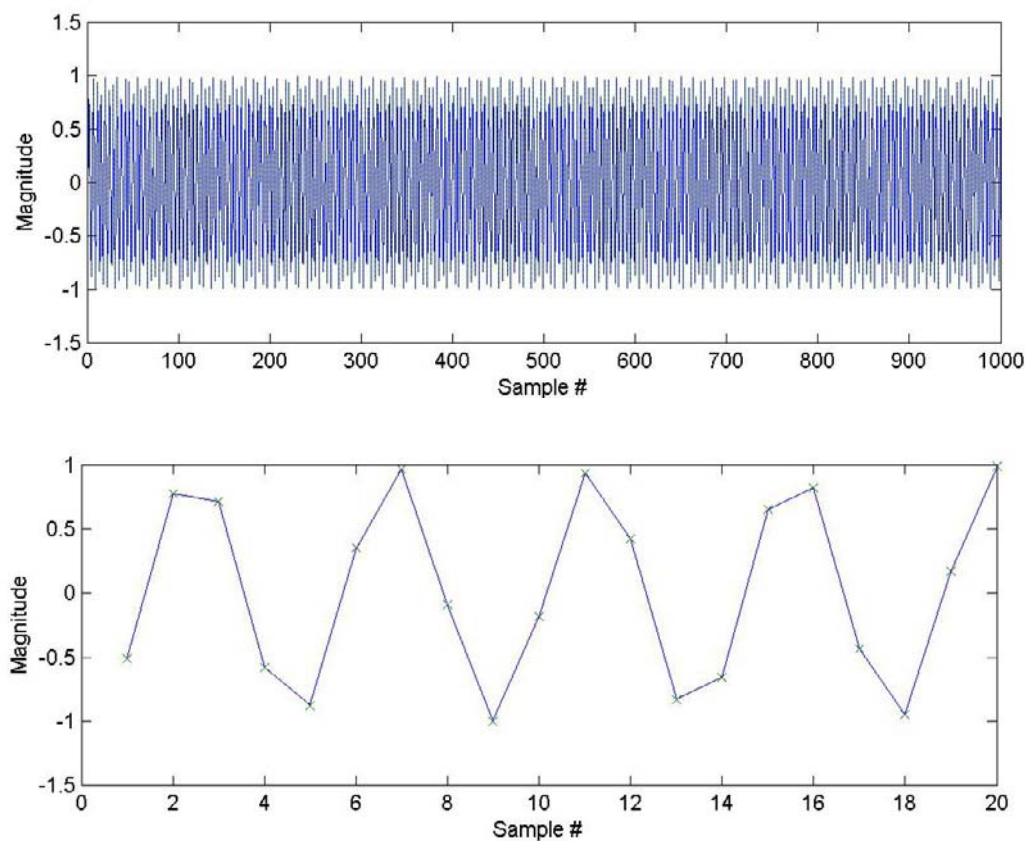
**Input signal power spectrum:**



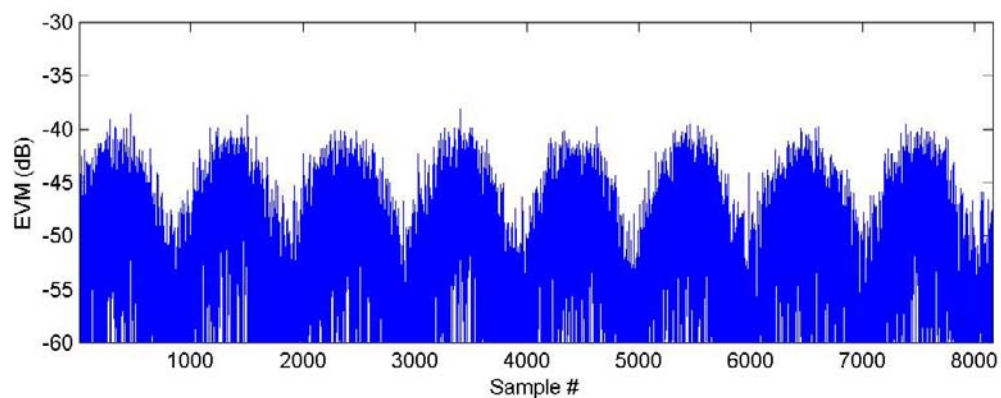


**Description:** Test of tone reconstruction, arbitrary frequency 22.857373 MHz.

**Typical time domain input vs output:**



**Typical EVM:**



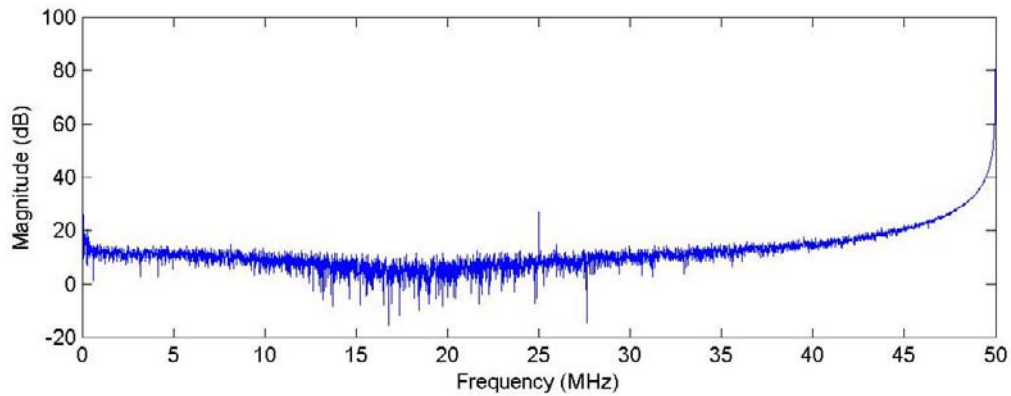
**Peak EVM:** -38.0dB



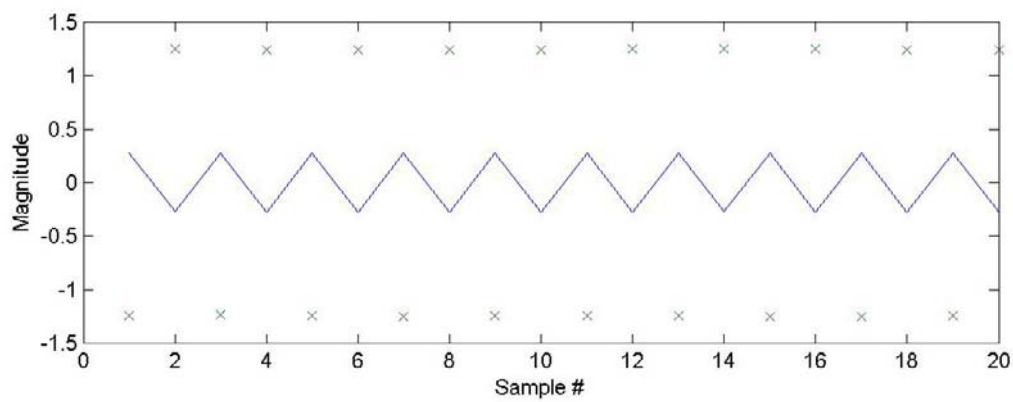
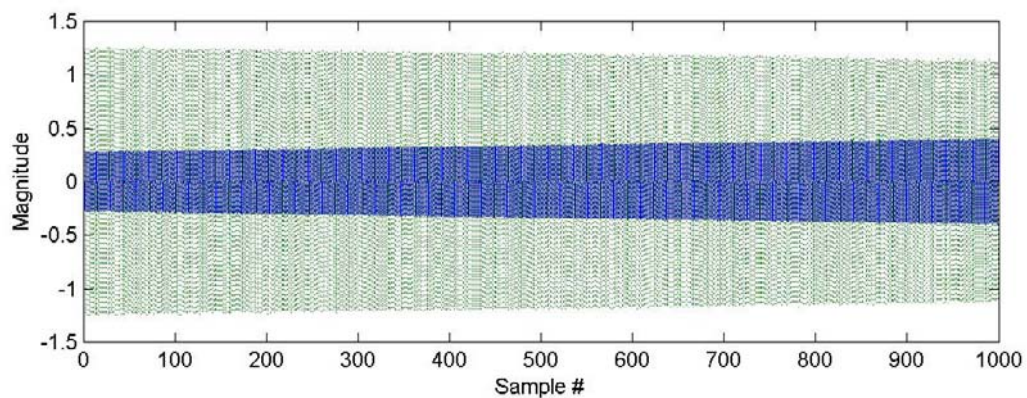
## B.12. 50 MHz Tone

**Description:** Test of tone reconstruction, high frequency 50 MHz.

**Input signal power spectrum:**

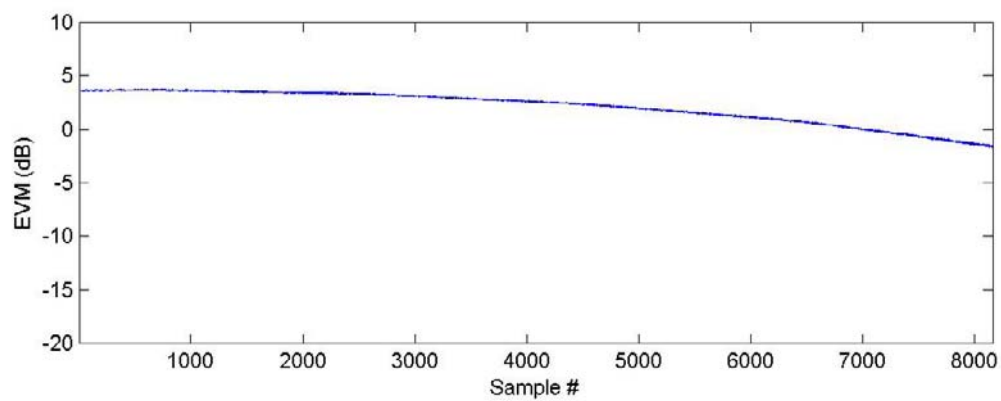


**Typical time domain input vs output:**



**Description:** Test of tone reconstruction, high frequency 50 MHz.

**Typical EVM:**

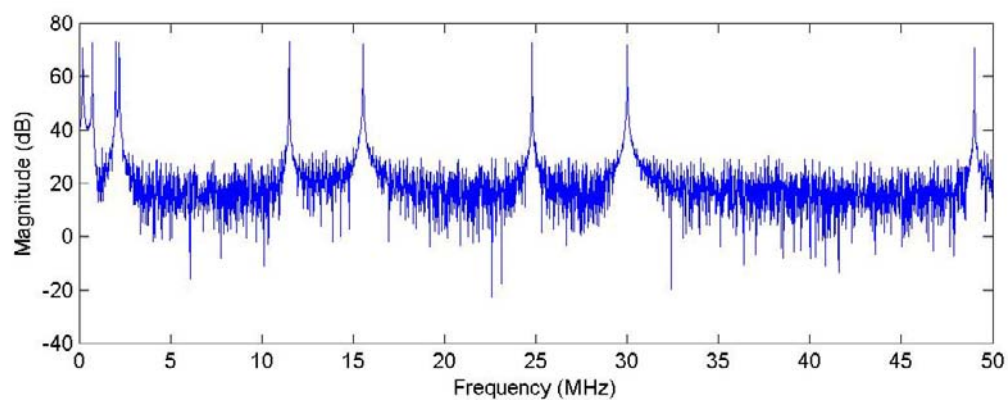


**Peak EVM:** +5.6dB

### B.13. Multi-tone Case#001

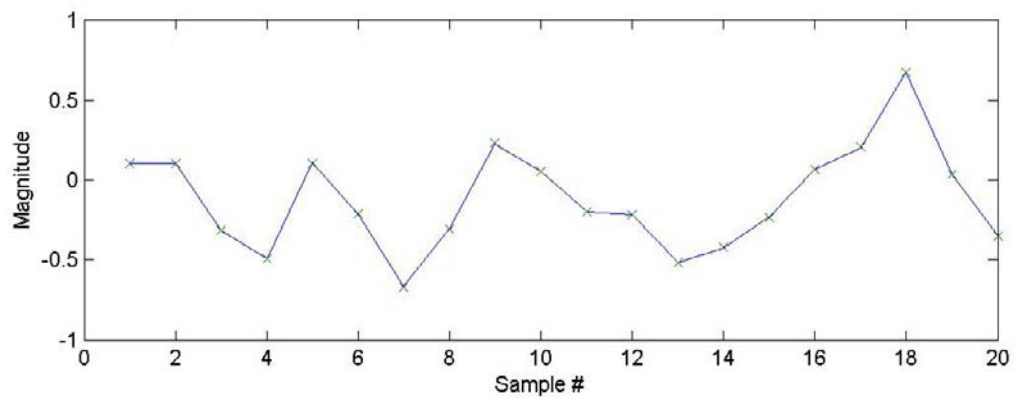
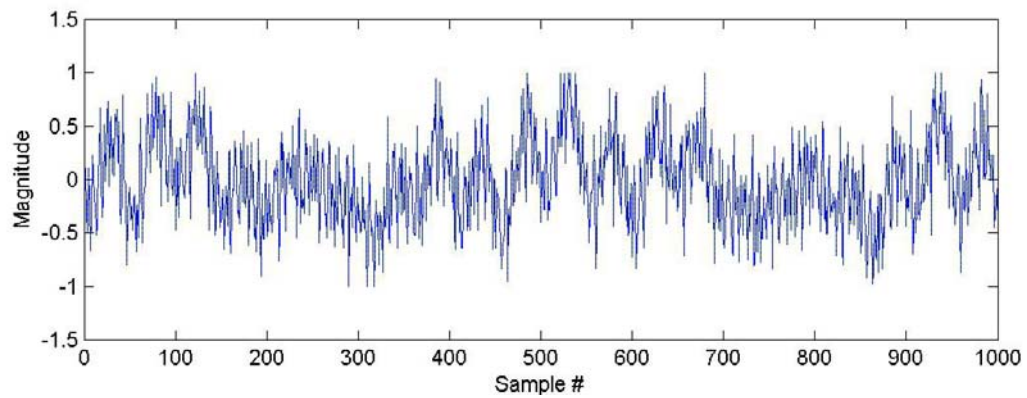
**Description:** Test of multi-tone signal reconstruction, case 1 of 4.

**Input signal power spectrum:**

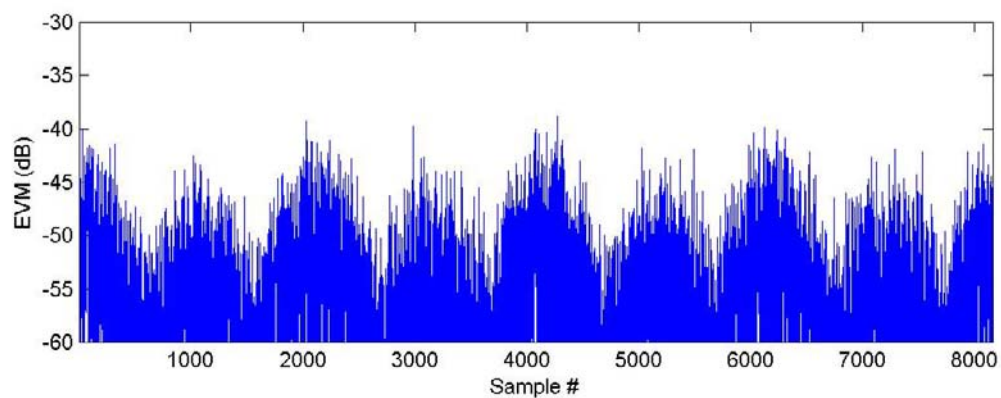


**Description:** Test of multi-tone signal reconstruction, case 1 of 4.

**Typical time domain input vs output:**



**Typical EVM:**

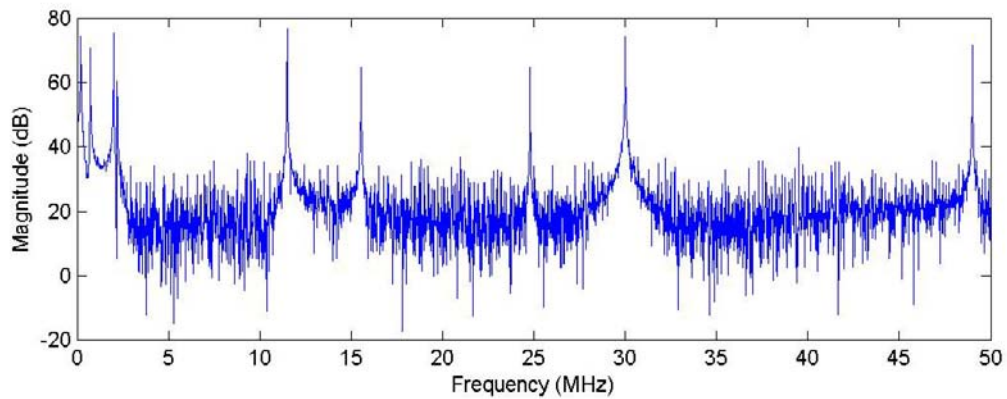


**Peak EVM:** -38.8dB

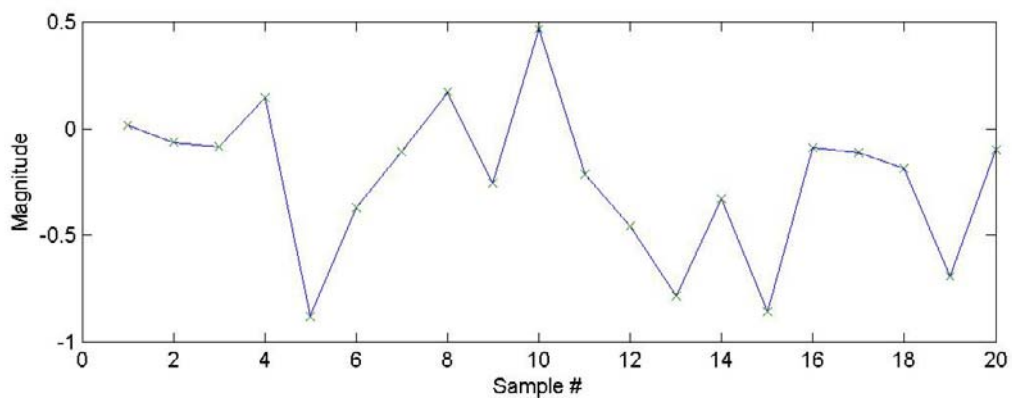
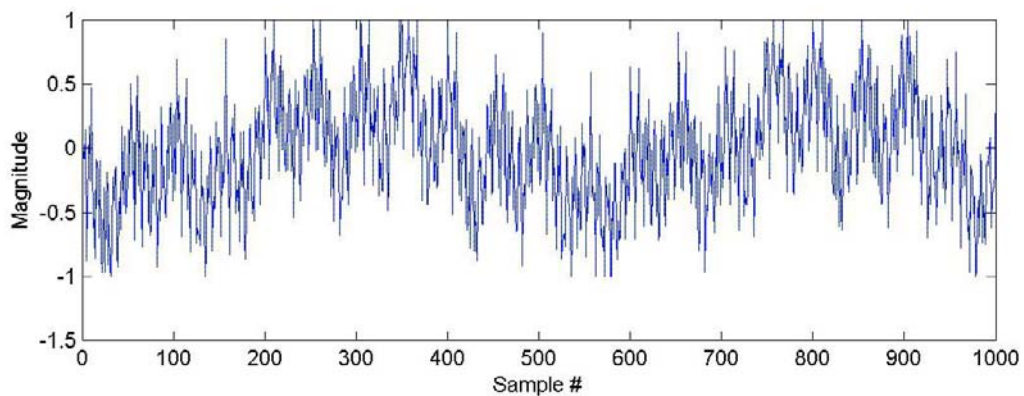
**B.14. Multi-tone Case#002**

**Description:** Test of multi-tone signal reconstruction, case 2 of 4.

**Input signal power spectrum:**

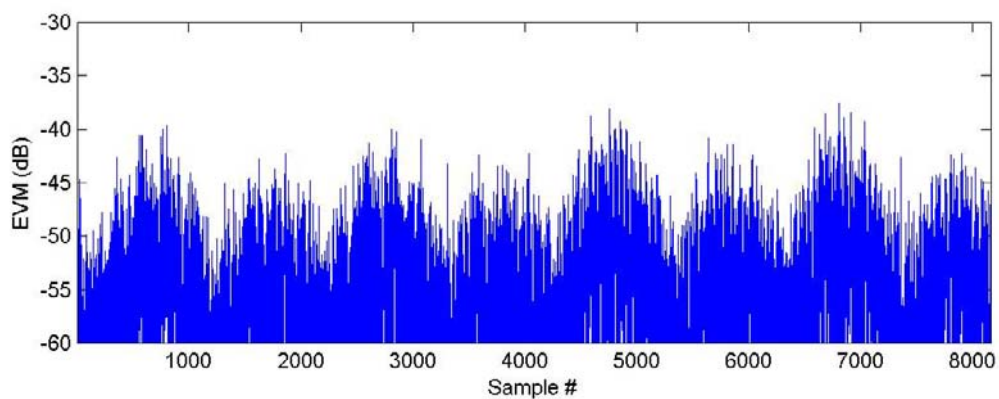


**Typical time domain input vs output:**



**Description:** Test of multi-tone signal reconstruction, case 2 of 4.

**Typical EVM:**

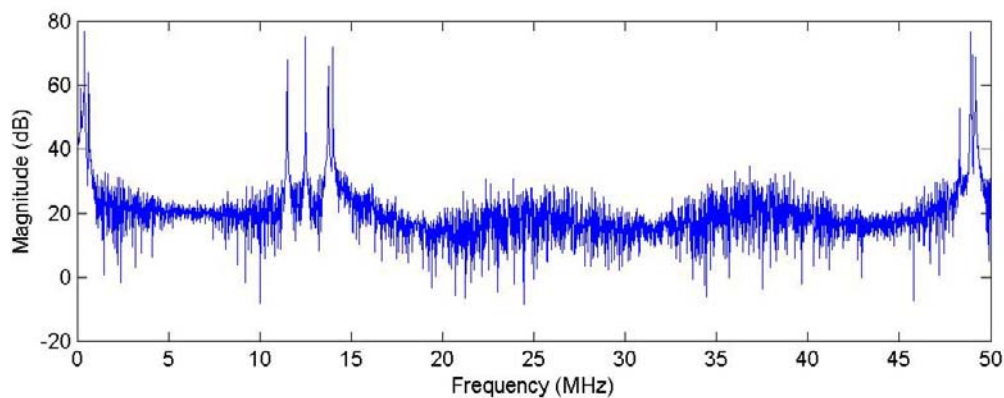


**Peak EVM:** -37.5dB

### B.15. Multi-tone Case#003

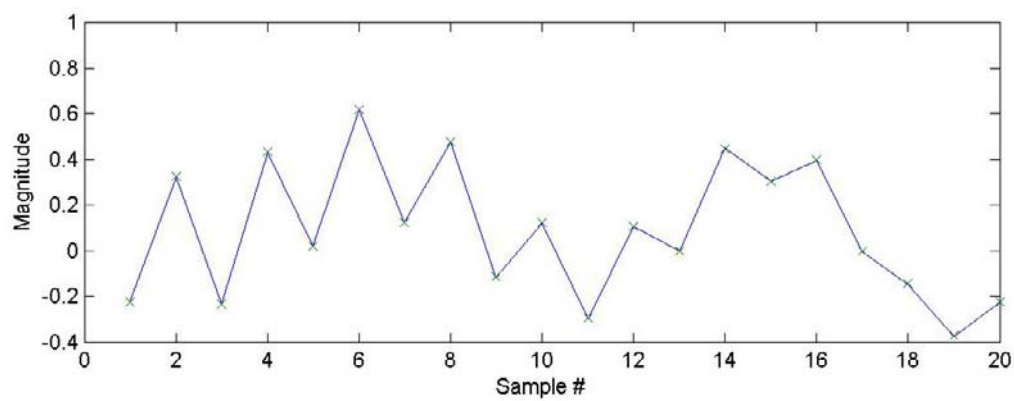
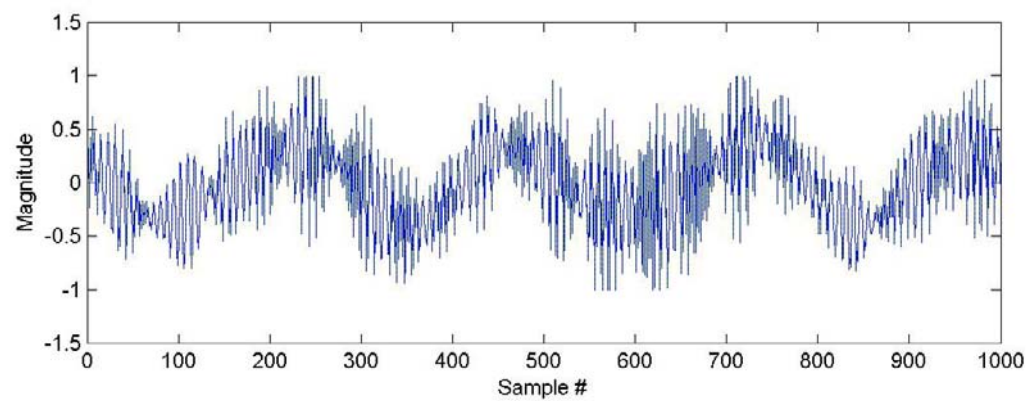
**Description:** Test of multi-tone signal reconstruction, case 3 of 4.

**Input signal power spectrum:**

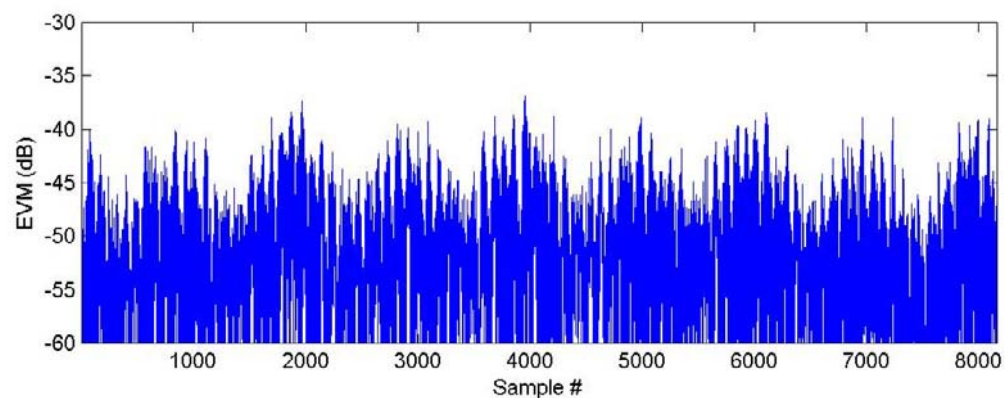


**Description:** Test of multi-tone signal reconstruction, case 3 of 4.

**Typical time domain input vs output:**



**Typical EVM:**



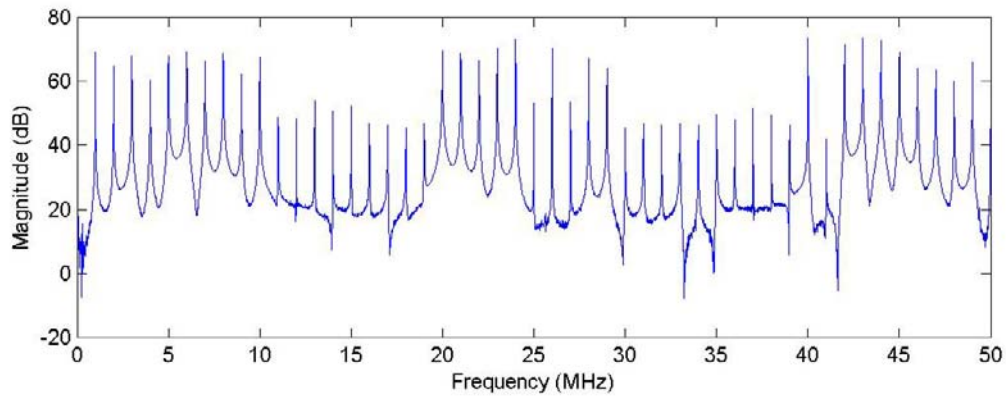
**Peak EVM:** -36.9dB



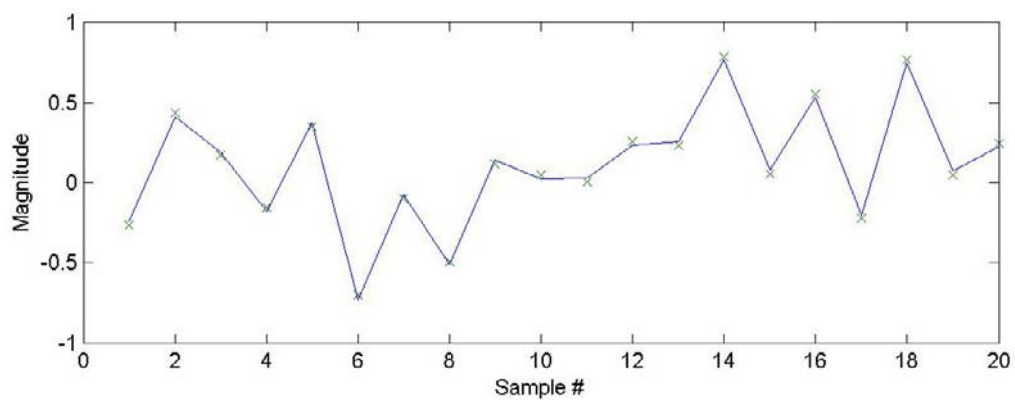
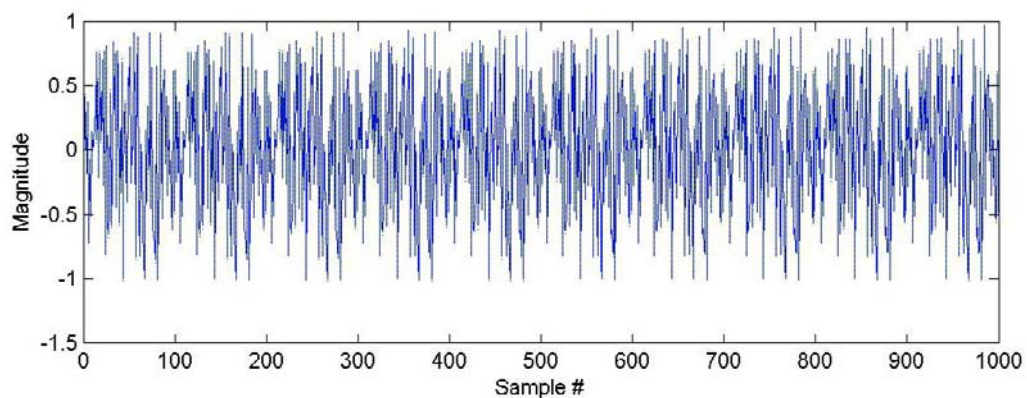
**B.16. Multi-tone Case#004**

**Description:** Test of multi-tone signal reconstruction, case 4 of 4.

**Input signal power spectrum:**



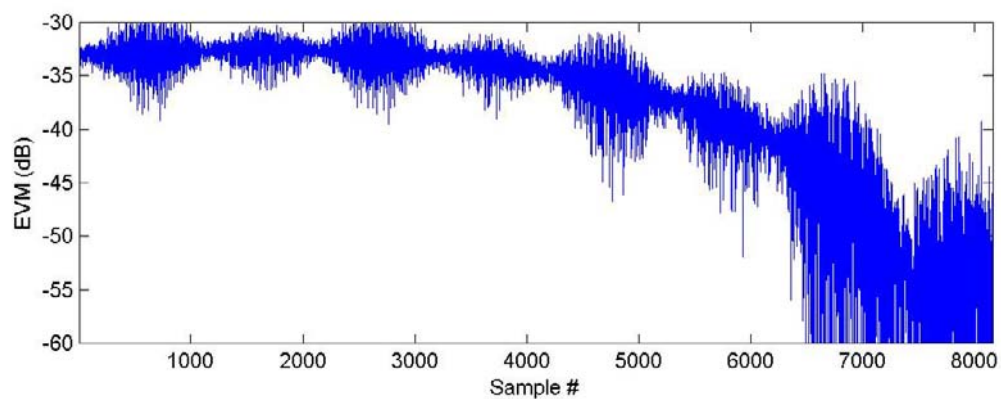
**Typical time domain input vs output:**





**Description:** Test of multi-tone signal reconstruction, case 4 of 4.

**Typical EVM:**

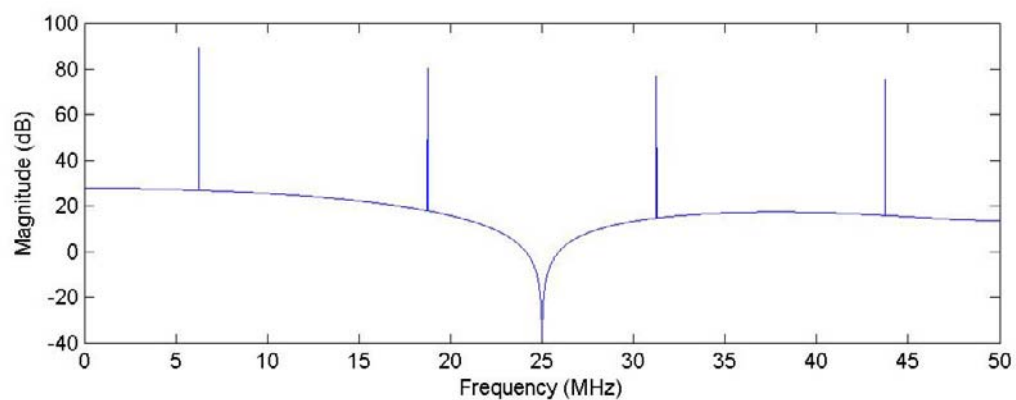


**Peak EVM:** -28.2dB

### B.17. Stored Waveform – Square Wave

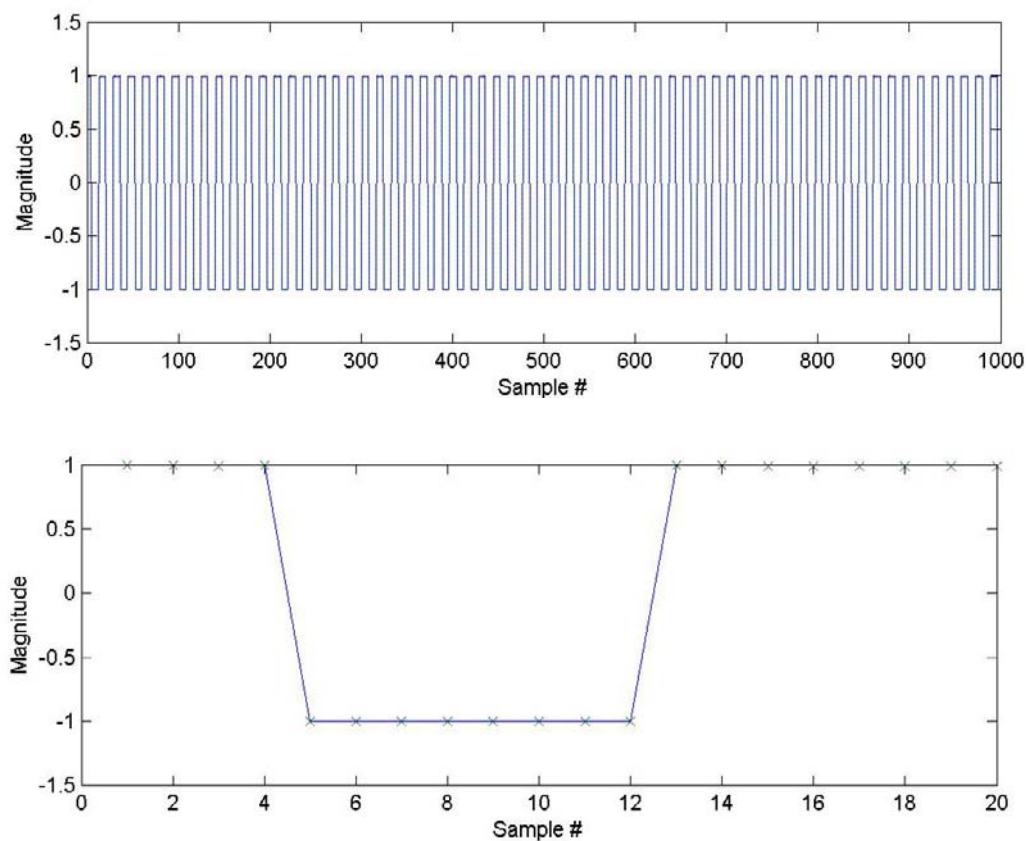
**Description:** Test of broadband signal reconstruction, square wave.

**Input signal power spectrum:**

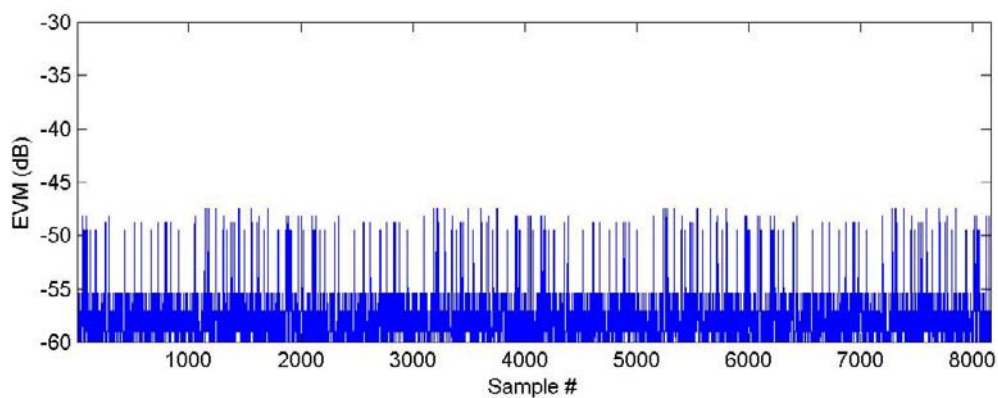


**Description:** Test of broadband signal reconstruction, square wave.

**Typical time domain input vs output:**



**Typical EVM:**

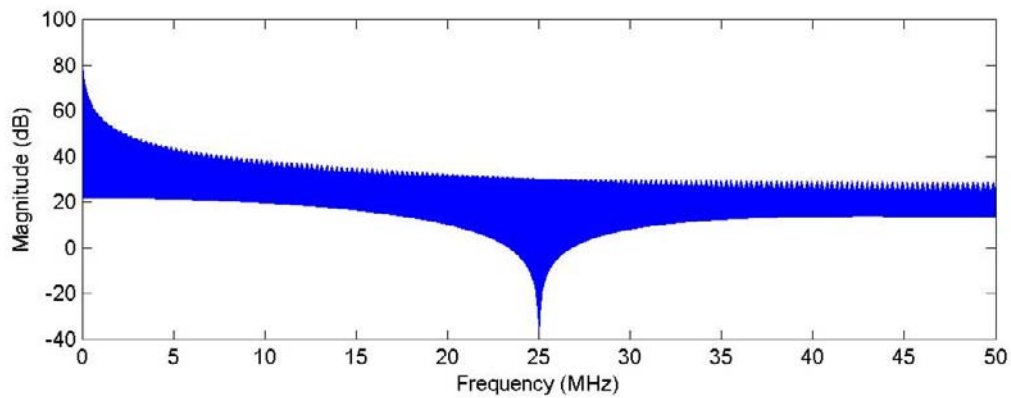


**Peak EVM:** -44.6dB

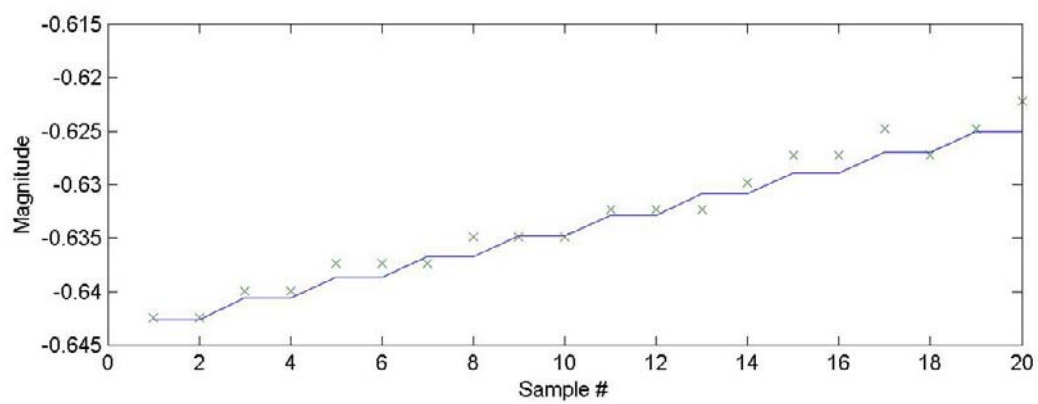
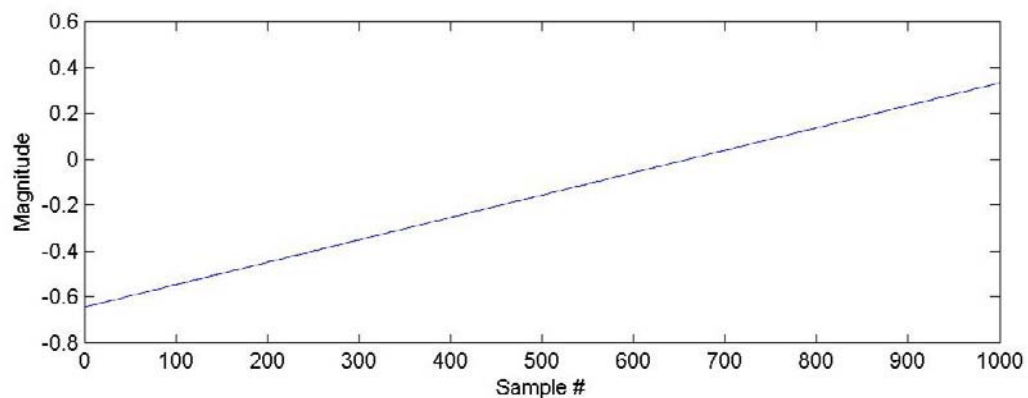
## B.18. Stored Waveform - Ramp

**Description:** Test of broadband signal reconstruction, ramp.

**Input signal power spectrum:**

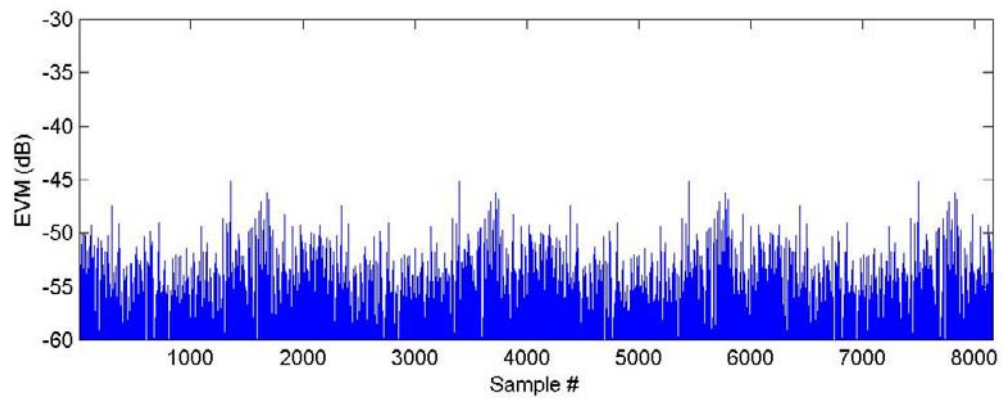


**Typical time domain input vs output:**



**Description:** Test of broadband signal reconstruction, ramp.

**Typical EVM:**



**Peak EVM:** -45.1dB

## Appendix C: MATLAB SCRIPTS

### C.1. Error Vector Magnitude Measurement

```
%
%   MATLAB m-script: EVM_for_ATP.m
%
%   Measure the time-domain reconstruction error vector magnitude in
%   sampled data at the input
%   and output of an RFEL filterbank application.
%
clear all;
%
%   List of logged data sets...(100MSPS unless otherwise noted)
%
%   *****
%   512-bin
%   *****
%   evm_066      Stored P-random function, on I and Q.
%                Passes EVM, offset = -3104.
%   evm_011      100kHz external sine input on I and Q, 130mVpp.
%                Passes EVM, offset = -3104.
%   evm_012      500kHz external sine input on I and Q, 130mVpp.
%                Passes EVM, offset = -3104.
%   evm_013      597.6kHz external sine input on I and Q, 130mVpp.
%                Passes EVM, offset = -3104.
%   evm_014      403.4kHz external sine input on I and Q, 130mVpp.
%                Passes EVM, offset = -3104.
%   evm_015      1MHz external sine input on I and Q, 130mVpp.
%                Passes EVM, offset = -3106.
%   evm_017      5MHz external sine input on I and Q, 130mVpp.
%                Passes EVM, offset = -3104.
%   evm_020      10MHz external sine input on I and Q, 130mVpp.
%                Passes EVM, offset = -3104.
%   evm_021      25MHz external sine input on I and Q, 130mVpp.
%                Passes EVM, offset = -3104.
%   evm_022      25MHz, external sine input on I and Q, 130mVpp.
%                Passes EVM, offset = -3106.
%   evm_023      40MHz external sine input on I and Q, 130mVpp.
%                Passes EVM, offset = -3104.
%   evm_072      50.00MHz external sine input on I and Q, 130mVpp.
%                Passes EVM, offset = -3104.
%   evm_033      MT001, direct from SMU I_Out and Q_Out (rear) ports, with
10dB pads.
%                Passes EVM, offset = -3106.
%   evm_035      MT002, direct from SMU I_Out and Q_Out (rear) ports, with
10dB pads.
%                Passes EVM, offset = -3104.
%   evm_037      MT003, direct from SMU I_Out and Q_Out (rear) ports, with
10dB pads.
%                Passes EVM, offset = -3104.
%   evm_065      MT004, direct from SMU I_Out and Q_Out (rear) ports, with
10dB pads.
%                Passes EVM, offset = -3104.
%   evm_030      Stored SQW function, on I and Q.
```

```

%           Passes EVM, offset = -3104.
%   evm_029   Stored ramp function, on I and Q.
%           Passes EVM, offset = -3104.
%   evm_042   Wideband AWGN from the SMU200A rear I/Q ports, driven by
AWGN only, 80MHz BW,10dB pads,
%           200MSPS.
%           Passes EVM, offset = -3106.
%*****
%   4096-bin
%*****
%   hr_evm_final_019   SMU AWGN
%           Fails EVM, offset = 0.
%   hr_evm_final_007   200kHz. Sine, 130mVpp/50ohm, dual ARB33250 FG's.
%           Fails EVM, offset = 0.
%   hr_evm_final_008   206kHz. Sine, 130mVpp/50ohm, dual ARB33250 FG's.
%           Fails EVM, offset = 0.
%   hr_evm_final_009   212kHz. Sine, 130mVpp/50ohm, dual ARB33250 FG's.
%           Passes EVM, offset = 0.
%   hr_evm_final_010   218kHz. Sine, 130mVpp/50ohm, dual ARB33250 FG's.
%           Passes EVM, offset = 0.
%   hr_evm_final_011   224kHz. Sine, 130mVpp/50ohm, dual ARB33250 FG's.
%           Fails EVM, offset = -2.
%   hr_evm_final_012   1MHz. Sine, 130mVpp/50ohm, dual ARB33250 FG's.
%           Passes EVM, offset = 0.
%   hr_evm_final_013   2MHz. Sine, 130mVpp/50ohm, dual ARB33250 FG's.
%           Passes EVM, offset = 0.
%   hr_evm_final_014   5MHz. Sine, 130mVpp/50ohm, dual ARB33250 FG's.
%           Fails EVM, offset = 0.
%   hr_evm_final_015   10MHz. Sine, 130mVpp/50ohm, dual ARB33250 FG's.
%           Fails EVM, offset = 0.
%   hr_evm_final_016   20MHz. Sine, 130mVpp/50ohm, dual ARB33250 FG's.
%           Fails EVM, offset = 0.
%   hr_evm_final_017   50MHz. Sine, 130mVpp/50ohm, dual ARB33250 FG's.
%           Fails EVM, offset = 0.
%   hr_evm_final_018   22.857373MHz. Sine, 130mVpp/50ohm, dual ARB33250
FG's.
%           Fails EVM, offset = 0.
%   hr_evm_final_003   MT001. Stored multi-tone from SMU200A, rear I/Q
with 10dB pads in-line.
%           Fails EVM, offset = 0.
%   hr_evm_final_004   MT002. Stored multi-tone from SMU200A, rear I/Q
with 10dB pads in-line.
%           Fails EVM, offset = 0.
%   hr_evm_final_005   MT003. Stored multi-tone from SMU200A, rear I/Q
with 10dB pads in-line.
%           Fails EVM, offset = 0.
%   hr_evm_final_006   MT004. Stored multi-tone from SMU200A, rear I/Q
with 10dB pads in-line.
%           Fails EVM, offset = 0.
%   hr_evm_final_002   Stored SQW function.
%           Passes EVM, offset = 0.
%   hr_evm_final_001   Stored RAMP function.
%           Passes EVM, offset = 0.
%*****
%*****

```

```

%
%   Specify the dataset relative PATH...
%
load datasets_lowres_v1_3\evm_043.txt -ascii;
% load datasets_hires_v1_3\hr_evm_final_006.txt -ascii;
%
dlog = evm_043;
%
%   Specify the offset...
%
offset = -3104;
ioffset = offset;
qoffset = offset;
%
% Extract the I and Q input and output channels...
%
iIn = dlog(:,2);
qIn = dlog(:,3);
iOut = dlog(:,4);
qOut = dlog(:,5);
L = length(iIn);
%
%   Copy the raw inputs for plotting...
%
Iiraw = iIn;
Ioraw = iOut;
Qiraw = qIn;
Qoraw = qOut;
%
%   Truncate datasets to eliminate non-overlapping segments that result
%   from capture latencies...
%
if ioffset >= 0
    iInChopped = iIn(ioffset+1:L);
    iOutChopped = iOut(1:L - ioffset);
else
    iInChopped = iIn(1:L+ioffset);
    iOutChopped = iOut(1-ioffset:L);
end;
%
if qoffset >= 0
    qInChopped = qIn(qoffset+1:L);
    qOutChopped = qOut(1:L - qoffset);
else
    qInChopped = qIn(1:L+qoffset);
    qOutChopped = qOut(1-qoffset:L);
end;
Lich = length(iInChopped);
Lqch = length(qInChopped);
Lch = min([Lich Lqch]);
%
% Normalise the input and output...
%
maxIIn = max(abs(iInChopped(1:Lch-10)));
maxIOut = max(abs(iOutChopped(1:Lch-10)));
maxQIn = max(abs(qInChopped(1:Lch-10)));
maxQOut = max(abs(qOutChopped(1:Lch-10)));

```



```

iOutChopped_n = iOutChopped ./ maxIOut;
iInChopped_n = iInChopped ./ maxIIn;
qOutChopped_n = qOutChopped ./ maxQOut;
qInChopped_n = qInChopped ./ maxQIn;

iO      =      iOutChopped_n      *      (std(iInChopped_n(1:Lich-10)))      /
std(iOutChopped_n(1:Lich-10)));
qO      =      qOutChopped_n      *      (std(qInChopped_n(1:Lqch-10)))      /
std(qOutChopped_n(1:Lqch-10)));
%
% Get the difference between the input and output signals...
%
diffI = 20 * log10(abs(iO - iInChopped_n));
diffQ = 20 * log10(abs(qO - qInChopped_n));
%
% Plot the results...
%
figure(1);
subplot(2,2,1), plot(Iiraw);
title('I Input');
subplot(2,2,2), plot(Ioraw);
title('I Output');
subplot(2,2,3), plot(Qiraw);
title('Q Input');
subplot(2,2,4), plot(Qoraw);
title('Q Output');
%
figure(2);
subplot(2,2,1), plot(1:L, iIn, 1:L, iOut);
title('Iin vs Iout (RAW)');
subplot(2,2,2), plot(1:Lich, iInChopped_n, 1:Lich, iO);
title('Iin vs Iout (TRUNCATED & SCALED)');
subplot(2,2,3), plot(1:L, qIn, 1:L, qOut);
title('Qin vs Qout (RAW)');
subplot(2,2,4), plot(1:Lqch, qInChopped_n, 1:Lqch, qO);
title('Qin vs Qout (TRUNCATED & SCALED)');
%
figure(3);
subplot(2,1,1), plot(diffI);
subplot(2,1,2), plot(diffQ);
max(diffI(2:Lch-30))
max(diffQ(2:Lch-30))
%
%   Plots a result for the ATP...
%
test = '4096b_pn_ext';
%
figure(10);
y = fft(iIn);
psd = y.*conj(y)/length(iIn);
f = (1e8/L).*(0:L/2);
plot(f/1e6, 10*log10(psd(1:(L/2+1))));
xlabel('Frequency (MHz)');
ylabel('Magnitude (dB)');
set(gca, 'PlotBoxAspectRatio', [2 0.7 1])%

```

```

% saveas(gca, strcat('C:\kev\Work Data\rfel_filterbank\psd_',test),
'jpg');
%
figure(11);
pl = 1000;
plot(1:pl, iInChopped_n(1:pl), 1:pl, iO(1:pl),':');
xlabel('Sample #');
ylabel('Magnitude');
set(gca,'PlotBoxAspectRatio',[2 0.7 1])%
% saveas(gca, strcat('C:\kev\Work Data\rfel_filterbank\zio_',test),
'jpg');
%
figure(12);
pl = 20;
plot(1:pl, iInChopped_n(1:pl), 1:pl, iO(1:pl),'x');
xlabel('Sample #');
ylabel('Magnitude');
set(gca,'PlotBoxAspectRatio',[2 0.7 1])%
% saveas(gca, strcat('C:\kev\Work Data\rfel_filterbank\zio_zoom_',test),
'jpg');
%
figure(13);
pl = Lch-30;
plot(1:pl, diffI(1:pl));
xlabel('Sample #');
ylabel('EVM (dB)');
set(gca,'PlotBoxAspectRatio',[2 0.7 1])%
axis([1 Lch-30 -60 -30]);
% saveas(gca, strcat('C:\kev\Work Data\rfel_filterbank\evm_',test),
'jpg');

```

## C.2. Test Result Log and Analysis

```

%
% MATLAB m-script: fbank_excision_tests.m
%
% Log and display the test results from the application of the
filterbank to narrowband
% interference excision with the DSSS demodulator.
%
clear all;
%
Gp = 17; %dB
%
%
*****
*****
% 4096-bin F/B
%
*****
*****
% 1. Degradation of DSSS BER in the presence of single-tone narrowband
interference
% as a function of relative jammer power and band position.

```

```

%
*****
*****
EbNo_4_f_2040 = [-30, 8.4e-9; -9, 9.2e-7; -6, 1.8e-5; -3, 4.9e-4; 0,
8.0e-3; 3, 8.5e-2];
EbNo_4_f_2045 = [-30, 8.4e-9; -9, 4.2e-7; -6, 6.3e-6; -3, 1.9e-4; 0,
3.2e-3; 3, 2.3e-2];
EbNo_4_f_2060 = [-30, 8.4e-9; -9, 5.6e-8; -6, 4.3e-7; -3, 6.9e-6; 0,
1.6e-4; 3, 2.7e-3; 6, 2.2e-2];
EbNo_4_f_2080 = [-30, 8.4e-9; -9, 1.3e-8; 9, 1.6e-8; 15, 3.0e-6; 21,
1.1e-3; 27, 2.0e-2];

EbNo_4_f_2040_exc = [3, 6.9e-7; 6, 2.2e-8; 15, 2.0e-7; 27, 1.4e-3; 30,
8.6e-3];
EbNo_4_f_2045_exc = [3, 7.4e-7; 6, 4.6e-8; 15, 1.3e-6; 21, 9.0e-6; 27,
2.5e-4; 33, 3.0e-2];
EbNo_4_f_2060_exc = [6, 3.2e-8; 15, 8.7e-8; 21, 7.9e-6; 27, 9.1e-5; 33,
2.0e-2];
EbNo_4_f_2080_exc = [27, 6.1e-7; 33, 1.5e-4; 39, 1.5e-2];

EbNo_2_f_2040 = [-30, 1.8e-7; -9, 5.4e-6; -6, 5.3e-5; -3, 7.8e-4; 0,
9.5e-3; 3, 2.0e-1];
EbNo_2_f_2045 = [-30, 1.8e-7; -9, 3.0e-6; -6, 2.4e-5; -3, 3.0e-4; 0,
3.4e-3; 3, 2.2e-2];
EbNo_2_f_2060 = [-30, 1.8e-7; -9, 7.4e-7; -6, 3.1e-6; -3, 2.3e-5; 0,
2.7e-4; 3, 2.9e-3; 6, 2.3e-2];
EbNo_2_f_2080 = [-30, 1.8e-7; -9, 1.4e-7; 15, 1.9e-5; 21, 1.5e-3; 27,
1.3e-2];

EbNo_2_f_2040_exc = [3, 2.0e-5; 6, 6.5e-7; 15, 9.9e-7; 27, 1.3e-3; 33,
2.0e-2];
EbNo_2_f_2045_exc = [3, 3.5e-5; 6, 1.0e-6; 15, 4.2e-6; 27, 6.4e-3; 30,
3.7e-2; 33, 1.0e-1];
EbNo_2_f_2060_exc = [6, 1.2e-6; 15, 4.0e-7; 27, 4.0e-4; 33, 1.1e-2];
EbNo_2_f_2080_exc = [27, 3.6e-6; 30, 2.6e-5; 33, 2.2e-4; 36, 2.5e-3; 39,
1.5e-2];

EbNo_0_f_2040 = [-30, 5e-6; -9, 3.8e-5; -6, 1.6e-4; -3, 1.1e-3; 0, 8.3e-
3; 3, 9.4e-2];
EbNo_0_f_2045 = [-30, 5e-6; -9, 3.1e-5; -6, 1.1e-4; -3, 7.0e-4; 0, 4.5e-
3; 3, 2.2e-2; 6, 1.0e-1];
EbNo_0_f_2060 = [-30, 5e-6; -9, 1.5e-5; -6, 3.3e-5; -3, 1.3e-4; 0, 7.4e-
4; 3, 5.0e-3; 6, 2.4e-2];
EbNo_0_f_2080 = [-30, 5e-6; -9, 5.5e-6; -6, 5.5e-6; -3, 5.7e-6; 0, 6.0e-
6; 3, 6.0e-6; 15, 9.3e-5;...
18, 5.0e-4; 24, 7.7e-3; 27, 2.0e-2];

EbNo_0_f_2040_exc = [6, 8.4e-6; 15, 2.9e-5; 27, 8.4e-3; 33, 6.6e-2];
EbNo_0_f_2045_exc = [6, 4.7e-5; 15, 5.8e-5; 27, 1.7e-2; 30, 5.3e-2];
EbNo_0_f_2060_exc = [6, 3.3e-5; 15, 1.4e-5; 27, 2.8e-3; 30, 1.7e-2];
EbNo_0_f_2080_exc = [27, 1.7e-4; 33, 7.1e-3; 36, 2.5e-2];

EbNo_n2_f_2040 = [-30, 1.4e-4; -9, 4.0e-4; -6, 9.2e-4; -3, 3.1e-3; 0,
1.3e-2; 3, 5.1e-2];
EbNo_n2_f_2045 = [-30, 1.4e-4; -9, 3.6e-4; -6, 7.8e-4; -3, 2.4e-3; 0,
9.0e-3; 3, 3.2e-2];

```

```

EbNo_n2_f_2060 = [-30, 1.4e-4; -9, 2.3e-4; -6, 3.5e-4; -3, 7.3e-4; 0,
2.0e-3; 3, 7.4e-3; 6, 2.7e-2];
EbNo_n2_f_2080 = [-30, 1.4e-4; -9, 1.8e-4; 0, 1.8e-4; 9, 1.8e-4; 15,
5.7e-4; 27, 2.0e-2];

EbNo_n2_f_2040_exc = [3, 3.6e-4; 15, 2.5e-4; 27, 5.2e-3; 30, 1.8e-2];
EbNo_n2_f_2045_exc = [3, 4.0e-4; 15, 3.4e-4; 27, 1.1e-2];
EbNo_n2_f_2060_exc = [6, 3.2e-4; 15, 2.4e-4; 27, 5.8e-3; 30, 2.1e-2];
EbNo_n2_f_2080_exc = [27, 8.0e-4; 30, 2.8e-3; 33, 1.1e-2];

EbNo_n4_f_2040 = [-30, 1.5e-3; -9, 2.6e-3; -6, 4.0e-3; -3, 8.1e-3; 0,
2.0e-2; 3, 5.6e-2];
EbNo_n4_f_2045 = [-30, 1.5e-3; -9, 2.4e-3; -6, 3.6e-3; -3, 7.0e-3; 0,
1.7e-2; 3, 4.3e-2];
EbNo_n4_f_2060 = [-30, 1.5e-3; -9, 1.9e-3; -6, 2.3e-3; -3, 3.6e-3; 0,
6.7e-3; 3, 1.5e-2];
EbNo_n4_f_2080 = [-30, 1.5e-3; -9, 1.5e-3; 9, 1.7e-3; 15, 2.9e-3; 27,
2.6e-2];

EbNo_n4_f_2040_exc = [3, 2.5e-3; 15, 2.4e-3; 27, 2.1e-2];
EbNo_n4_f_2045_exc = [3, 2.5e-3; 15, 2.0e-3; 27, 1.0e-2];
EbNo_n4_f_2060_exc = [3, 2.3e-3; 15, 1.7e-3; 27, 5.8e-3; 30, 1.1e-2];
EbNo_n4_f_2080_exc = [27, 3.0e-3; 33, 1.1e-2];

% figure(1);
% hold on;
% plot(Gp-EbNo_4_f_2040(:,1), EbNo_4_f_2040(:,2), 'k');
% plot(Gp-EbNo_4_f_2045(:,1), EbNo_4_f_2045(:,2), 'k');
% plot(Gp-EbNo_4_f_2060(:,1), EbNo_4_f_2060(:,2), 'k');
% plot(Gp-EbNo_4_f_2080(:,1), EbNo_4_f_2080(:,2), 'k');
%
% plot(Gp-EbNo_4_f_2040_exc(:,1), EbNo_4_f_2040_exc(:,2), 'k');
% plot(Gp-EbNo_4_f_2045_exc(:,1), EbNo_4_f_2045_exc(:,2), 'k');
% plot(Gp-EbNo_4_f_2060_exc(:,1), EbNo_4_f_2060_exc(:,2), 'k');
% plot(Gp-EbNo_4_f_2080_exc(:,1), EbNo_4_f_2080_exc(:,2), 'k');
%
% plot(Gp-EbNo_2_f_2040(:,1), EbNo_2_f_2040(:,2), 'c');
% plot(Gp-EbNo_2_f_2045(:,1), EbNo_2_f_2045(:,2), 'c');
% plot(Gp-EbNo_2_f_2060(:,1), EbNo_2_f_2060(:,2), 'c');
% plot(Gp-EbNo_2_f_2080(:,1), EbNo_2_f_2080(:,2), 'c');
%
% plot(Gp-EbNo_2_f_2040_exc(:,1), EbNo_2_f_2040_exc(:,2), 'c');
% plot(Gp-EbNo_2_f_2045_exc(:,1), EbNo_2_f_2045_exc(:,2), 'c');
% plot(Gp-EbNo_2_f_2060_exc(:,1), EbNo_2_f_2060_exc(:,2), 'c');
% plot(Gp-EbNo_2_f_2080_exc(:,1), EbNo_2_f_2080_exc(:,2), 'c');
%
% plot(Gp-EbNo_0_f_2040(:,1), EbNo_0_f_2040(:,2), 'b');
% plot(Gp-EbNo_0_f_2045(:,1), EbNo_0_f_2045(:,2), 'b');
% plot(Gp-EbNo_0_f_2060(:,1), EbNo_0_f_2060(:,2), 'b');
% plot(Gp-EbNo_0_f_2080(:,1), EbNo_0_f_2080(:,2), 'b');
%
% plot(Gp-EbNo_0_f_2040_exc(:,1), EbNo_0_f_2040_exc(:,2), 'b');
% plot(Gp-EbNo_0_f_2045_exc(:,1), EbNo_0_f_2045_exc(:,2), 'b');
% plot(Gp-EbNo_0_f_2060_exc(:,1), EbNo_0_f_2060_exc(:,2), 'b');
% plot(Gp-EbNo_0_f_2080_exc(:,1), EbNo_0_f_2080_exc(:,2), 'b');
%
% plot(Gp-EbNo_n2_f_2040(:,1), EbNo_n2_f_2040(:,2), 'g');

```

```

% plot(Gp-EbNo_n2_f_2045(:,1), EbNo_n2_f_2045(:,2),'g');
% plot(Gp-EbNo_n2_f_2060(:,1), EbNo_n2_f_2060(:,2),'g');
% plot(Gp-EbNo_n2_f_2080(:,1), EbNo_n2_f_2080(:,2),'g');
%
% plot(Gp-EbNo_n2_f_2040_exc(:,1), EbNo_n2_f_2040_exc(:,2), 'g');
% plot(Gp-EbNo_n2_f_2045_exc(:,1), EbNo_n2_f_2045_exc(:,2), 'g');
% plot(Gp-EbNo_n2_f_2060_exc(:,1), EbNo_n2_f_2060_exc(:,2), 'g');
% plot(Gp-EbNo_n2_f_2080_exc(:,1), EbNo_n2_f_2080_exc(:,2), 'g');
%
% plot(Gp-EbNo_n4_f_2040(:,1), EbNo_n4_f_2040(:,2),'r');
% plot(Gp-EbNo_n4_f_2045(:,1), EbNo_n4_f_2045(:,2),'r');
% plot(Gp-EbNo_n4_f_2060(:,1), EbNo_n4_f_2060(:,2),'r');
% plot(Gp-EbNo_n4_f_2080(:,1), EbNo_n4_f_2080(:,2),'r');
%
% plot(Gp-EbNo_n4_f_2040_exc(:,1), EbNo_n4_f_2040_exc(:,2), 'r');
% plot(Gp-EbNo_n4_f_2045_exc(:,1), EbNo_n4_f_2045_exc(:,2), 'r');
% plot(Gp-EbNo_n4_f_2060_exc(:,1), EbNo_n4_f_2060_exc(:,2), 'r');
% plot(Gp-EbNo_n4_f_2080_exc(:,1), EbNo_n4_f_2080_exc(:,2), 'r');
% %
% set( gca,'YScale', 'log');
% axis([-20 50 1e-9 1e-1]);
% hold off;
% grid on;
% ylabel('BER')
% xlabel(' (P/J)*(W/R) (dB) ');
% title('Excision-aided BER performance versus single tone jammer power &
frequency, for a set of Eb/No');

%
%
*****
*****
% 2. Degradation of DSSS BER in the presence of multi-tone narrowband
interference
% as a function of relative jammer power and number of tones.
%
*****
*****
EbNo_4_n32x1 = [-30, 1.1e-8; -9, 2.2e-7; -6, 2.8e-6; -3, 1.4e-4; 0, 3.2e-
3; 3, 2.4e-2];
EbNo_0_n32x1 = [-30, 4.8e-6; -15, 7.9e-6; -9, 2.7e-5; -6, 9.9e-5; -3,
5.2e-4; 0, 3.7e-3; 3, 2.1e-2];
EbNo_n4_n32x1 =[-30, 1.5e-3; -15, 1.7e-3; -9, 2.5e-3; -6, 3.5e-3; -3,
6.8e-3; 0, 1.5e-2; 3, 3.9e-2];

EbNo_4_n32x1_exc =[3, 8.0e-6; 12, 9.8e-6; 18, 4.6e-4; 21, 2.8e-3; 24,
1.7e-3; 30, 3.0e-2];
EbNo_0_n32x1_exc = [3, 3.5e-4; 6, 2.7e-4; 9, 1.4e-4; 21, 4.7e-3; 27,
9.7e-3; 30, 2.7e-2];
EbNo_n4_n32x1_exc =[3, 5.1e-3; 15, 6.8e-3; 21, 1.5e-2; 24, 1.9e-2; 30,
4.5e-2];

% figure(2);
% hold on;
% plot(Gp-EbNo_4_n32x1(:,1), EbNo_4_n32x1(:,2),'g');
% plot(Gp-EbNo_0_n32x1(:,1), EbNo_0_n32x1(:,2),'b');
% plot(Gp-EbNo_n4_n32x1(:,1), EbNo_n4_n32x1(:,2),'r');

```

```

% plot(Gp-EbNo_4_f_2040(:,1), EbNo_4_f_2040(:,2), 'gx');
% plot(Gp-EbNo_0_f_2040(:,1), EbNo_0_f_2040(:,2), 'bx');
% plot(Gp-EbNo_n4_f_2040(:,1), EbNo_n4_f_2040(:,2), 'rx');

% plot(Gp-EbNo_4_n32x1_exc(:,1), EbNo_4_n32x1_exc(:,2), 'g');
% plot(Gp-EbNo_0_n32x1_exc(:,1), EbNo_0_n32x1_exc(:,2), 'b');
% plot(Gp-EbNo_n4_n32x1_exc(:,1), EbNo_n4_n32x1_exc(:,2), 'r');

% plot(Gp-EbNo_4_f_2040_exc(:,1), EbNo_4_f_2040_exc(:,2), 'go');
% plot(Gp-EbNo_0_f_2040_exc(:,1), EbNo_0_f_2040_exc(:,2), 'bo');
% plot(Gp-EbNo_n4_f_2040_exc(:,1), EbNo_n4_f_2040_exc(:,2), 'ro');

%
% set( gca, 'YScale', 'log');
% axis([-20 50 1e-9 1e-1]);
% hold off;
% grid on;
% ylabel('BER')
% xlabel(' (P/J)*(W/R) (dB) ');
% title('Excision-aided BER performance versus multi-tone jammer power,
for a set of Eb/No');

%
*****
*****
% 512-bin F/B
%
*****
*****
% 1. Degradation of DSSS BER in the presence of single-tone narrowband
interference
% as a function of relative jammer power and band position.
%
*****
*****
%
EbNo_512b_0_f_2040 = [-30, 5.7e-6; -9, 4.4e-5; -6, 1.9e-4; -3, 1.3e-3; 0,
8.7e-3; 3, 1.1e-1];
EbNo_512b_0_f_2040_exc = [3, 1.6e-5; 9, 1.6e-5; 15, 2.7e-5; 27, 7.1e-3;
30, 2.8e-2];
%
%
*****
*****
% 2. Degradation of DSSS BER in the presence of multi-tone narrowband
interference
% as a function of relative jammer power and number of tones.
%
*****
*****
%
EbNo_512b_0_n32x1 = [-30, 6.3e-6; -15, 9.4e-6; -9, 2.9e-5; -6, 9.9e-5; -
3, 4.8e-4; 0, 3.4e-3; 3, 1.9e-2; 6, 9.8e-2];
EbNo_512b_0_n32x1_exc = [6, 3.3e-2];
%
figure(1);

```

```

hold on;
plot(Gp-EbNo_512b_0_f_2040(:,1), EbNo_512b_0_f_2040(:,2), 'k:');
plot(Gp-EbNo_512b_0_f_2040_exc(:,1), EbNo_512b_0_f_2040_exc(:,2), 'k');
plot(Gp-EbNo_512b_0_n32x1(:,1), EbNo_512b_0_n32x1(:,2), 'b:');
plot(Gp-EbNo_512b_0_n32x1_exc(:,1), EbNo_512b_0_n32x1_exc(:,2), 'bx');
set( gca, 'YScale', 'log');
axis([-20 50 1e-9 1e-1]);
hold off;
grid on;
ylabel('BER')
xlabel(' (P/J)*(W/R) (dB) ');
title('512-bin filterbank excision performance');

```



<b>DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION</b>							
<b>DOCUMENT CONTROL DATA</b>						1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)	
2. TITLE  Wideband High-Resolution Filterbank Firmware: Acceptance Testing and Excision Filtering (U)				3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)  Document (U) Title (U) Abstract (U)			
4. AUTHOR(S)  K. L. Harman and C. Potter				5. CORPORATE AUTHOR  DSTO Defence Science and Technology Organisation PO Box 1500 Edinburgh South Australia 5111 Australia			
6a. DSTO NUMBER DSTO-TR-2092		6b. AR NUMBER AR- 014-084		6c. TYPE OF REPORT Technical Report		7. DOCUMENT DATE January 2008	
8. FILE NUMBER 2007/1101174		9. TASK NUMBER INT 07/020		10. TASK SPONSOR ASCP		11. NO. OF PAGES 101	
						12. NO. OF REFERENCES 12	
13. URL on the World Wide Web  <a href="http://www.dsto.defence.gov.au/corporate/reports/DSTO-TR-2092.pdf">http://www.dsto.defence.gov.au/corporate/reports/DSTO-TR-2092.pdf</a>					14. RELEASE AUTHORITY  Chief, Command, Control, Communications and Intelligence Division		
15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT  <i>Approved for public release</i>  OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111							
16. DELIBERATE ANNOUNCEMENT  No Limitations							
17. CITATION IN OTHER DOCUMENTS No							
18. DSTO RESEARCH LIBRARY THESAURUS <a href="http://web-vic.dsto.defence.gov.au/workareas/library/resources/dsto_thesaurus.htm">http://web-vic.dsto.defence.gov.au/workareas/library/resources/dsto_thesaurus.htm</a>  Performance Evaluation, Spectral Analysis, Filters, Digital Signal processing							
19. ABSTRACT A filterbank is a signal processing tool that can facilitate manipulation of signals in the frequency domain. Under contract to DSTO, RF Engines Limited in the United Kingdom developed a 512-bin and a 4096-bin filterbank as firmware cores, capable of real-time digital processing at complex sample rates of 200MSPS (390kHz per bin) and 100MSPS (24kHz per bin) respectively. These cores will facilitate research and development into high-performance adaptive filtering in support of Defence wireless communications tasks. In this document, the integration of the cores into an existing platform and subsequent acceptance testing are reported, confirming that the cores do satisfy their design specification. Then the application of the cores to front-end excision filtering in a direct-sequence spread-spectrum receiver is investigated, and demonstrates a highly effective performance enhancement.							